MODE HUNTING AND DENSITY ESTIMATION WITH THE FOCUSSED INFORMATION CRITERION

by

EMIL KVERNELAND MOGSTAD

THESIS
for the degree of
MASTER OF SCIENCE

(Master i Modellering og Dataanalyse)

Faculty of Mathematics and Natural Sciences
University of Oslo

May 2013

Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo

Preface

I want to thank my supervisor, Nils Lid Hjort, who has made me work independently on the thesis, but still maintaining the right amount of control to get the thesis ashore in time. Nils has given me the right pushes throughout the process, something which has made me take longer steps into the theory, and learn more, than I believed that I could.

I would also like to thank my supportive family, my fellow students and friends, and especially Realistforeningen[1], for being an inclusive social environment with kind people through my years at the university.

I would also like to thank my girlfriend Katja, for her care and understanding.

Oslo, May 2013
Emil Mogstad

[1] Direct translation: The Scientist Union. A social student organization for mathematics and natural science students.

Contents

Thesis Overview

The idea behind this project was to construct a general scheme for mode hunting using the Focussed Information Criterion, and a family of models defined by

$$f_m(y;\xi,\sigma,\boldsymbol{a}) = \phi(\epsilon)\frac{1}{\sigma}\exp\left(\sum_{j=1}^{m} a_j\psi_j(\Phi(\epsilon))\right)\frac{1}{k_m(\boldsymbol{a})},$$

where $\epsilon = (y-\xi)/\sigma$,$\psi_k(u) = \sqrt{2}\cos(k\pi u)$ and ϕ, Φ are the normal density and cumulative distribution functions respectively. Later in the process, density estimation was included in the project, where the model was selected with Average-FIC. The opponent to the scheme is kernel density estimation, introduced py Parzen (1962).

The thesis starts with an introductory chapter, which builds all the theory used in the rest of the thesis. This includes maximum likelihood estimaton with asymptotic normality, and derivation of the FIC and Average-FIC, with the steps explained along the way. The last section is dedicated to kernel density estimation, and discussions about optimal smoothing for mode hunting, and density estimation.

In chapter two, the scheme is introduced. It starts with a brief explorations of the ψ_k function family, and then goes over to maximum likelihood. In order to compute FIC, a number of vectors and matrices are needed, which are derived and explained in this chapter. At the end of chapter two, a step by step example of how to use the scheme is given, if the reader would wish to expore the scheme them self.

Chapter three starts with the introduction of a test distribution family, commonly known as the normal mixture, with density function

$$g(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p}) = \sum_{i=1}^{k} p_i\phi\left(\frac{y-\mu_i}{\tau_i}\right)\frac{1}{\tau_i}, \quad \sum_{i=1}^{n} p_i = 1.$$

While kernel estimation is the opponent, this has the role of a reference distribution. The rest of the chapter is dedicated to the mode hunt. Both analytical and simulation based approaches for comparing the kernel estimate to the parametric estimate are given and discussed.

Chapter four is similar to chapter three. The same reference distributions are used for the tests, however this chapter deals with density estimation.

A lot of Python scripts has been written for this thesis to do various computations and simulations. A short guide in how to use them is found in appendix

C.1. For a more comprehensive documentation, visit
`http://folk.uio.no/emilkm/scriptsdoc/`.

Guide to Notation

Some notation is standard for the entire thesis. This list gathers most of them. Vectors are generally noted with bold faced characters. Theorems, lemmas, corollaries and propositions are placed in boxes, while proofs are ended with a ■, and examples with a □.

Ω	Sample space, which in this thesis are subsets of $\mathbb{R}$.
$\boldsymbol{\theta}$	General vector of parameters. Denotes parameters belonging to the narrow model when related to FIC.
$\boldsymbol{\gamma}$	The extension of the parameter vector from narrow to wide model.
$\boldsymbol{\gamma}_0$	The choice of $\boldsymbol{\gamma}$, such that the wide model becomes the narrow one.
ω	Related to FIC, $\omega = J_{10}J_{00}^{-1}\frac{\partial \mu}{\partial \boldsymbol{\theta}} - \frac{\partial \mu}{\partial \boldsymbol{\gamma}}$
$\mathcal{L}_n(\cdot), \ell_n(\cdot)$	Likelihood function, and log likelihood function
$\ell(\cdot)$	The log density function, $\ell(\cdot) = \log f(y)$.
S_n	Partial derivatives of the log likelihood function with respect to the parameters.
S	Partial derivatives of the log density function with respect to the parameters.
D_{KL}	The Kullback-Leibler divergence
$I(\cdot)$	Indicator function. $I = 1$ if the argument is true.
μ	Focus parameter.
J, J_n	Hessian of the log density and log likelihood functions respectively.
$\mathcal{O}(\cdot)$	Big-O notation, $f(n) = \mathcal{O}(g(n))$ if f and g are asymptotically proportional.
$\phi(\cdot)$	The standard normal density function.
$\Phi(\cdot)$	The standard normal cumulative distribution function.
$g(y)$	Target distribution/true data generating density function.
$y_0, \hat{y}_0$	Mode of distribution.
$\hat{y}_{0,K}, \hat{y}_{0,P}$	Kernel and parametric estimates of the mode.
$\to_P, \to_D$	Convergence in probability, and convergence in distribution.
$=_D$	$X_n =_D Y_n$ indicates that X_n and Y_n has the same limiting distribution.

Chapter 1

Introduction

1.1 The Likelihood Principle

Assume a sample of n independent and identically distributed random variables $y_1, \ldots, y_n$, with common density function $f(y; \boldsymbol{\theta})$. The likelihood function, and log likelihood function are defined as

$$\mathcal{L}_n(\boldsymbol{y}; \boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i; \boldsymbol{\theta})$$

$$\ell_n(\boldsymbol{y}; \boldsymbol{\theta}) = \log\left(\prod_{i=1}^{n} f(y_i; \boldsymbol{\theta})\right) = \sum_{i=1}^{n} \log f(y_i; \boldsymbol{\theta}).$$

The parameter vector $\hat{\boldsymbol{\theta}}_n$ which maximizes $\mathcal{L}_n(\boldsymbol{y}; \boldsymbol{\theta})$, is the maximum likelihood estimate of $\boldsymbol{\theta}$. It is desireable to use the log-likelihood function $\ell_n(\boldsymbol{y}; \boldsymbol{\theta})$ instead, for numerical stability and mathematical convenience.

Example 1.1.1 (The Exponential Distribution) *Assume a random sample $y_1, \ldots, y_n$ of independent random variables from the exponential distribution, with log likelihood function*

$$\ell_n(\boldsymbol{y}; \lambda) = \sum_{i=1}^{n} [\log \lambda - \lambda y_i] = n \log \lambda - \lambda \sum_{i=1}^{n} y_i,$$

and score function

$$S_n(\boldsymbol{y}; \lambda) = \frac{n}{\lambda} - \sum_{i=1}^{n} y_i.$$

Solving the equation $S_n = 0$ for λ gives that $\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} y_i} = \frac{1}{\bar{y}}$.

□

1.1.1 The Kullback-Leibler Divergence

Akaike (1973) discusses the link between maximum likelihood estimation and the Kullback-Leibler divergence, defined as

$$D_{KL}\left(g \parallel f\right) = \int_{-\infty}^{\infty} \log \frac{g(y)}{f(y;\boldsymbol{\theta})} g(y) dy.$$

The measure D_{KL} is non-negative, and equals 0 if and only if $f = g$. A rewrite gives

$$\int_{-\infty}^{\infty} g(y) \left(\log g(y) - \log f(y;\boldsymbol{\theta})\right) dy. \tag{1.1}$$

Assume that we want to estimate a density $g(y)$ with model candidate $f(y;\boldsymbol{\theta})$ based on a random sample $Y_1, \ldots, Y_n$. The first term of (1.1) is equal for every parameter $\boldsymbol{\theta}$, so minimizing the Kullback-Leibler divergence is equivalent to maximizing

$$\int_{-\infty}^{\infty} g(y) \log f(y;\boldsymbol{\theta}) dy.$$

By the law of large numbers, we have that

$$\frac{1}{n}\ell_n(\boldsymbol{y};\boldsymbol{\theta}) \to_P E_g[\log f(y;\boldsymbol{\theta})] = \int_{-\infty}^{\infty} g(y) \log f(y;\boldsymbol{\theta}) dy$$

for all $\boldsymbol{\theta} \in \Omega$, provided that the integral exist. Given existence and uniqueness of the minimizer $\boldsymbol{\theta}_0$ of $E_g[\log f(y;\boldsymbol{\theta})]$, we have that

$$\hat{\boldsymbol{\theta}}_n \to_P \boldsymbol{\theta}_0 = \arg\min_{\boldsymbol{\theta}} \{D_{KL}(g \parallel f(y;\boldsymbol{\theta}))\},$$

where $\boldsymbol{\theta}_0$ is called the least false parameter.

Example 1.1.2 (Estimating Gamma with the Exponential Distribution) *Assume the gamma distribution with density function*

$$g(y;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta y}.$$

Setting $\alpha = 1$ gives the exponential distribution with parameter β. The KL divergence from the gamma to the exponential distribution is

$$\begin{aligned} D_{KL} &= \int_0^\infty \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y} \log\left(\frac{\beta^\alpha}{\lambda\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y - (-\lambda y)}\right) dy \\ &\propto \int_0^\infty y^{\alpha-1} e^{-\beta y} \left(\alpha \log\beta - \log\lambda - \log\Gamma(\alpha) + (\alpha-1)\log y - (\beta-\lambda)y)\right) dy. \end{aligned}$$

Differentiating the integral with respect to λ, gives the equation

$$\int_0^\infty y^{\alpha-1}e^{-\beta y}\left(-\frac{1}{\lambda}+y\right)dy = -\frac{1}{\lambda}\frac{\Gamma(\alpha)}{\beta^\alpha}+\frac{\Gamma(\alpha+1)}{\beta^{\alpha+1}}=0,$$

which has solution

$$\lambda = \frac{\beta\Gamma(\alpha)}{\Gamma(\alpha+1)} = \frac{\beta}{\alpha} = \frac{1}{E[Y]}.$$

This shows that the least false parameter $\lambda_0 = \frac{1}{E[Y]}$ is the unique minimizer of the Kullback-Leibler divergence from the gamma to the exponential distribution. Since the sample based estimator for $1/E[Y]$ is $1/\bar{y}$, this is consistent with example 1.1.1.

□

1.2 Maximum Likelihood Asymptotics

Recall that, given a sample of independent random variables $y_1, \ldots, y_n$, with common distribution $f(y;\boldsymbol{\theta})$, then the log likelihood function is

$$\ell_n(\boldsymbol{y};\boldsymbol{\theta}) = \log\left(\prod_{i=1}^n f(y_i;\boldsymbol{\theta})\right) = \sum_{i=1}^n \log(f(y_i;\boldsymbol{\theta})).$$

Define also the log density function

$$\ell(y;\boldsymbol{\theta}) = \log\left[f(y;\boldsymbol{\theta})\right],$$

and note that $\ell'(y;\boldsymbol{\theta})$ is the vector of partial derivatives of $\ell(y;\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, while $\ell''(y;\boldsymbol{\theta})$ is the corresponding hessian.

This chapter is taken from Knight (2000), and restated here since it plays an important part of this thesis. Assume that f satisfies

- c1 The parameter space Θ is an open subset of $\mathbb{R}^p$
- c2 The set $\Omega = \{y : f(y;\boldsymbol{\theta}) > 0\}$ does not depend on $\boldsymbol{\theta}$
- c3 $f(y;\boldsymbol{\theta})$ is three times continously differentiable with respect to $\boldsymbol{\theta}$ for all $\boldsymbol{\theta} \in \Omega$
- c4 $E_f[\ell'(y;\boldsymbol{\theta})] = 0$ for all $\boldsymbol{\theta}$, and $\mathrm{cov}_f[\ell'(y;\boldsymbol{\theta})] = K(\boldsymbol{\theta})$, where $K(\boldsymbol{\theta})$ is positive definite for all $\boldsymbol{\theta}$.
- c5 $E_f[\ell''(y;\boldsymbol{\theta})] = -J(\boldsymbol{\theta})$, where $J(\boldsymbol{\theta})$ is positive definite for all $\boldsymbol{\theta}$
- c6 Let $\ell'''_{jkl}(y;\boldsymbol{\theta})$ be the mixed partial derivative of ℓ, with respect to θ_j, θ_k and θ_l. For each $\boldsymbol{\theta}, \delta > 0$, $|\ell'''_{jkl}(x;\boldsymbol{t})| \leq M_{jkl}(y)$ for $||\boldsymbol{\theta}-\boldsymbol{t}|| \leq \delta$ where $E_\theta[M_{jkl}(y)] < \infty$

From condition c2, we know that for all $\boldsymbol{\theta} \in \Theta$

$$\int_\Omega f(y;\boldsymbol{\theta})dy = 1, \tag{1.2}$$

and

$$\frac{\partial f}{\partial \boldsymbol{\theta}} \int_\Omega f(y;\boldsymbol{\theta})dy = 0.$$

Moving the derivative inside the integral gives

$$0 = \int_\Omega \frac{\partial f}{\partial \boldsymbol{\theta}} f(y;\boldsymbol{\theta})dy = \int_\Omega \ell'(y;\boldsymbol{\theta}) f(y;\boldsymbol{\theta})dy = E_\theta[\ell'(y_i;\boldsymbol{\theta})].$$

Differentiating once more gives that

$$\begin{aligned} 0 &= \int_\Omega \frac{\partial}{\partial \boldsymbol{\theta}} \ell'(y;\boldsymbol{\theta}) f(y;\boldsymbol{\theta})dy \\ &= \int_\Omega \frac{\partial \ell}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^t}(y;\boldsymbol{\theta}) f(y;\boldsymbol{\theta})dy + \int_\Omega \frac{\partial \ell}{\partial \boldsymbol{\theta}} \frac{\partial \ell}{\partial \boldsymbol{\theta}}^t f(y;\boldsymbol{\theta})dy \\ &= -J(\boldsymbol{\theta}) + K(\boldsymbol{\theta}) \end{aligned}$$

,
which gives that $J(\boldsymbol{\theta}) = K(\boldsymbol{\theta}) = \text{cov}[\ell'(y_i;\boldsymbol{\theta})]$. Further on, assume that

$$\sum_{i=1}^n \ell'(y_i;\hat{\boldsymbol{\theta}}_n) = 0,$$

which by a Taylor expansion about $\boldsymbol{\theta}$ gives

$$\begin{aligned} 0 &= \sum_{i=1}^n \ell'(y_i;\hat{\boldsymbol{\theta}}_n) \\ &= \sum_{i=1}^n \ell'(y_i;\boldsymbol{\theta}) + (\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) \sum_{i=1}^n \ell''(y_i;\hat{\boldsymbol{\theta}}_n) + \frac{1}{2}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta})^T \sum_{i=1}^n \ell'''(y_i;\boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}), \end{aligned}$$

where $\boldsymbol{\theta}^*$ lies somewhere between $\hat{\boldsymbol{\theta}}_n$ and $\boldsymbol{\theta}$. Multiplying the equation by $\sqrt{n}$ gives that

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) = \frac{-\frac{1}{\sqrt{n}} \sum_{i=1}^n \ell'(y_i;\boldsymbol{\theta})}{\frac{1}{n}\sum_{i=1}^n \ell''(y_i;\hat{\boldsymbol{\theta}}_n) + \frac{1}{2n}\sum_{i=1}^n \ell'''(y_i;\boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta})}. \tag{1.3}$$

From the central limit theorem, and condition c4, it follows that

$$\frac{1}{\sqrt{n}} \sum_{i=1}^n \ell'(y_i;\boldsymbol{\theta}) \to_D \mathcal{N}(0, K(\boldsymbol{\theta})),$$

and from condition c5, and the weak law of large numbers it follows that

$$\frac{1}{n}\sum_{i=1}^{n}\ell''(y_i;\hat{\boldsymbol{\theta}}_n) \to_P -J(\boldsymbol{\theta}).$$

Thus Slutsky's theorem we then have that

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) \to_D \mathcal{N}(0, J(\boldsymbol{\theta})^{-1}K(\boldsymbol{\theta})J(\boldsymbol{\theta})^{-1}) = \mathcal{N}(0, J(\boldsymbol{\theta})^{-1}),$$

provided that (proven in Knight (2000, p. 253))

$$\frac{1}{2n}\sum_{i=1}^{n}\ell'''(y_i;\boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) \to_P 0.$$

We are now ready to state the main theorem. First define

$$J(\boldsymbol{\theta}) = -\int_\Omega \frac{\partial^2 \ell}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^t} f(y;\boldsymbol{\theta})dy$$

$$K(\boldsymbol{\theta}) = \int_\Omega \frac{\partial \ell}{\partial \boldsymbol{\theta}}\frac{\partial \ell}{\partial \boldsymbol{\theta}}^t f(y;\boldsymbol{\theta})dy = \operatorname{cov}\left[\frac{\partial \ell}{\partial \boldsymbol{\theta}}\right].$$

Theorem 1.2.1 (Asymptotic Normality of MLEs) *Assume that observations $y_1, y_2, \ldots, y_n$ are independent and identically distributed with a distribution $f(y;\boldsymbol{\theta})$, which satisfies condition c1-c6, and assume that the MLE satisfy $\hat{\boldsymbol{\theta}}_n \to_p \boldsymbol{\theta}$ where*

$$\sum_{i=1}^{n}\frac{\partial \ell}{\partial \boldsymbol{\theta}}\ell(y_i,\hat{\boldsymbol{\theta}}_n) = \mathbf{0}$$

then

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) \to_D \mathcal{N}(0, J(\boldsymbol{\theta})^{-1}) \tag{1.4}$$

The asymptotic distribution derived above is done under the assumption that $f = g$ is the known true data generating process. In most realistic situations this is not the case, and we have no guarantee that $K = J$ holds. We know that the estimated parameter $\hat{\boldsymbol{\theta}}_n$ converges to the least false parameter $\boldsymbol{\theta}_0$, and Huber (1967) proved that

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \to_d \mathcal{N}(0, J(\boldsymbol{\theta}_0)^{-1}K(\boldsymbol{\theta}_0)J(\boldsymbol{\theta}_0)^{-1}). \tag{1.5}$$

This is consistent with theorem 1.2.1 if $K = J$. In order to estimate J and K, we use the sample estimates

$$\hat{J}(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2 \ell(y_i;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^t}$$

$$\hat{K}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial \ell(y_i;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\frac{\partial \ell(y_i;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^t$$

with the maximum likelihood estimate $\hat{\boldsymbol{\theta}}_n$ plugged in for $\boldsymbol{\theta}$.

1.2.1 Confidence Intervals

Assume independent observations $y_1, \ldots, y_n$, and a maximum likelihood estimate $\hat{\boldsymbol{\theta}}_n$ under a model f. We know that

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) \to_D \mathcal{N}(0, \Sigma)$$

where Σ is estimated as either $J(\hat{\boldsymbol{\theta}}_n)^{-1}$ or the sandwich estimate in (1.5). It can be shown that for parameter $\hat{\theta}_i \in \hat{\boldsymbol{\theta}}_n$, we have that

$$\sqrt{n}(\hat{\theta}_i - \theta_i) \to_D \mathcal{N}\left(0, \Sigma_{i,i}\right).$$

For a confidence interval, plug in the sample estimate $\hat{\Sigma}$ of Σ. We have that

$$CI(\hat{\theta}_i) = \hat{\theta}_i \pm \sqrt{\frac{\hat{\Sigma}_{i,i}}{n}} Z_{(1-\frac{\alpha}{2})}.$$

The latter can be used to check if $\hat{\theta}_i$ is significant or not. For this thesis, the asymptotic distribution of a focus parameters is needed. One tool to achieve this is the Δ-method [1]

Theorem 1.2.2 (The Δ-method) *Let $\boldsymbol{X}_n$ be a random vector and $\boldsymbol{a}$ a vector in $\mathbb{R}^p$ such that*

$$\sqrt{n}(\boldsymbol{X}_n - \boldsymbol{a}) \to_D \mathcal{N}_p\left(0, \Sigma\right).$$

If f is function $f : \mathbb{R}^p \to \mathbb{R}^q$, which is differentiable at $\boldsymbol{a}$, then

$$\sqrt{n}(f(\boldsymbol{X}_n) - f(\boldsymbol{a})) \to_D \mathcal{N}_p\left(0, \mathcal{J}^t \Sigma \mathcal{J}\right),$$

where $\mathcal{J}$ is the Jacobi matrix of f evaluated at $\boldsymbol{a}$.

For proof see Lehmann (1999, Thm 5.4.6).

Let $\mu : \mathbb{R}^p \to \mathbb{R}$ be a focus parameter. Then we have that the limiting distribution of $\hat{\mu} = \mu(\hat{\boldsymbol{\theta}})$ is

$$\sqrt{n}(\hat{\mu} - \mu) \to_D \mathcal{N}\left(0, \left(\frac{\partial \mu}{\partial \boldsymbol{\theta}}\right)^t \Sigma \frac{\partial \mu}{\partial \boldsymbol{\theta}}\right),$$

which gives that a two sided confidence interval for μ on level α is

$$CI(\hat{\mu}) = \hat{\mu} \pm Z_{(1-\frac{\alpha}{2})} \sqrt{\frac{1}{n}\left(\frac{\partial \mu}{\partial \hat{\boldsymbol{\theta}}}\right)^t \hat{\Sigma} \frac{\partial \mu}{\partial \hat{\boldsymbol{\theta}}}}.$$

[1] Δ is the greek letter capital "delta", so the theorem is often called the "delta-method".

1.3 Focus Parameters

In Claeskens and Hjort (2008, p. 119), the parameter distribution for a model with local misspesification is discussed, and forms the basis of the Focussed Information Criterion. Assume a random sample $y_1 \ldots, y_n$ of independent random variables from a sequence of distributions f_n

$$f_n(y) = f\left(y; \boldsymbol{\theta}, \gamma_0 + \frac{\delta}{\sqrt{n}}\right),$$

where $\boldsymbol{\theta}$ is a parameter vector in $\mathbb{R}^p$ and γ in $\mathbb{R}^q$. Let the wide model $f(y; \boldsymbol{\theta}, \boldsymbol{\gamma})$ include parameter γ, and let $\gamma = \gamma_0$ give the narrow model $f(y; \boldsymbol{\theta}_0)$ as a special case of the wide model.

The question is whether to include γ as a parameter. Leaving $\gamma = \gamma_0$ gives higher modelling bias, while estimating it may increase variance. Let $(\hat{\boldsymbol{\theta}}, \hat{\gamma})$ be the estimated parameters in the wide model, and $\hat{\boldsymbol{\theta}}_0$ the estimated parameter in the narrow model. Also let

$$J = \begin{pmatrix} J_{00} & J_{01} \\ J_{10} & J_{11} \end{pmatrix}, \qquad J^{-1} = \begin{pmatrix} J^{00} & J^{01} \\ J^{10} & J^{11} \end{pmatrix}$$

be the full $(p+q) \times (p+q)$ information matrix derived for the wide model, but calulated with $\gamma = \gamma_0$. From Claeskens and Hjort (2008, p. 122) we have that

Theorem 1.3.1 *As n goes to infinity, we have that*

$$\sqrt{n}\begin{pmatrix} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ (\hat{\gamma} - \gamma_0) \end{pmatrix} \to_D \mathcal{N}_{p+q}\left(\begin{pmatrix} 0 \\ \delta \end{pmatrix}, J^{-1}\right) \tag{1.6}$$

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}) \to_D \mathcal{N}_p\left(J_{00}^{-1} J_{01}\delta, J_{00}^{-1}\right) \tag{1.7}$$

Proof: For the first part, the wide maximum likelihood estimate will have an approximate distribution

$$\sqrt{n}\begin{pmatrix} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ (\hat{\gamma} - (\gamma_0 + \delta/\sqrt{n})) \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, J^{-1}\right)$$

which gives that

$$\sqrt{n}\begin{pmatrix} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ (\hat{\gamma} - \gamma_0) \end{pmatrix} - \begin{pmatrix} \mathbf{0} \\ \delta \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, J^{-1}\right)$$

which proves the statement. For the second part, similar reasoning as in (1.3) gives that

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}) =_D J_{00}^{-1}\sqrt{n}\bar{U}_n$$

where $\bar{U}$ is the partial derivative of the log density function for the narrow model. We know that $\sqrt{n}\bar{U}_n$ has an approximate normal distribution with covariance J_{00}. For the bias we see that by a Taylor expansion of f at γ_0 we get

$$f\left(y;\boldsymbol{\theta},\gamma_0+\frac{\delta}{\sqrt{n}}\right)=f(y;\boldsymbol{\theta},\gamma_0)+\left(\gamma_0+\frac{\delta}{\sqrt{n}}-\gamma_0\right)\frac{\partial f}{\partial\gamma_0}+o\left(\frac{1}{\sqrt{n}}\right) \tag{1.8}$$

$$\approx f\left(y;\boldsymbol{\theta},\gamma_0\right)\left(1+\frac{\delta}{\sqrt{n}}\frac{\partial\ell}{\partial\gamma_0}\right).$$

This gives that

$$E[\bar{U}_n]\approx\int_{-\infty}^{\infty}f(y;\boldsymbol{\theta},\gamma)\left(1+\frac{\delta}{\sqrt{n}}\frac{\partial\ell}{\partial\gamma}\right)U(y)dy$$

$$=\frac{\delta}{\sqrt{n}}E\left[\frac{\partial\ell}{\partial\gamma}\frac{\partial\ell}{\partial\boldsymbol{\theta}}\right]=J_{01}\frac{\delta}{\sqrt{n}},$$

which leads to the result

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_0-\boldsymbol{\theta})=_D J_{00}^{-1}\sqrt{n}\bar{U}_n\rightarrow_D\mathcal{N}\left(J_{00}^{-1}J_{01}\delta,J_{00}^{-1}\right).$$

■

Let $\mu(\boldsymbol{\theta},\gamma)$ be a focus parameter. The focus parameter can be any differentiable function related to a random variable, such as a quantile, the distribution mode, or its mean. For the following corollary we let

$$\hat{\mu}_{narr}=\mu(\hat{\boldsymbol{\theta}}_0,\gamma_0)$$

$$\hat{\mu}_{wide}=\mu(\hat{\boldsymbol{\theta}},\hat{\gamma}).$$

Corollary 1.3.1 *As n goes to infinity, we have that*

$$\sqrt{n}(\hat{\mu}_{narr}-\mu_{true})\rightarrow_D\mathcal{N}\left(\boldsymbol{\omega}^t\boldsymbol{\delta},\tau_0^2\right) \tag{1.9}$$

$$\sqrt{n}(\hat{\mu}_{wide}-\mu_{true})\rightarrow_D\mathcal{N}\left(0,\tau_0^2+\boldsymbol{\omega}^tQ\boldsymbol{\omega}\right) \tag{1.10}$$

where $\omega=J_{10}J_{00}^{-1}\frac{\partial\mu}{\partial\boldsymbol{\theta}}-\frac{\partial\mu}{\partial\gamma}$, $\tau_0^2=\left(\frac{\partial\mu}{\partial\boldsymbol{\theta}}\right)^tJ_{00}^{-1}\frac{\partial\mu}{\partial\boldsymbol{\theta}}$ *with derivatives taken at* $(\boldsymbol{\theta}_0,\gamma_0)$ *and* $Q=J^{11}$.

Proof: We see by the Δ-method that

$$\sqrt{n}(\hat{\mu}_{wide} - \mu_{true}) =_D \begin{pmatrix} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \end{pmatrix}^t \sqrt{n} \begin{pmatrix} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ (\hat{\boldsymbol{\gamma}} - (\boldsymbol{\gamma}_0 + \boldsymbol{\delta}/\sqrt{n})) \end{pmatrix},$$

which has an approximate normal distribution with zero mean and variance

$$\tau^2 = \begin{pmatrix} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \end{pmatrix}^t J_{wide}^{-1} \begin{pmatrix} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \end{pmatrix}.$$

Let $Q = \left[J_{11} - J_{10} J_{00}^{-1} J_{01} \right]^{-1}$, be the lower right block of J^{-1}, and use that

$$J^{10} = -J_{00}^{-1} J_{01} Q, \qquad J^{00} = J_{00}^{-1} + J_{00}^{-1} J_{01} Q J_{10} J_{00}^{-1}.$$

Then

$$\begin{aligned}
\tau^2 &= \frac{\partial \mu}{\partial \boldsymbol{\theta}}^t \left(J_{00}^{-1} + J_{00}^{-1} J_{01} Q J_{10} J_{00}^{-1} \right) \frac{\partial \mu}{\partial \boldsymbol{\theta}} - 2 \frac{\partial \mu}{\partial \boldsymbol{\theta}}^t J_{00}^{-1} J_{01} Q \frac{\partial \mu}{\partial \boldsymbol{\gamma}} + \frac{\partial \mu}{\partial \boldsymbol{\gamma}}^t Q \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \\
&= \tau_0^2 + \left(J_{10} J_{00}^{-1} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \right)^t Q \left(J_{10} J_{00}^{-1} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \right) - 2 \left(J_{10} J_{00}^{-1} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \right)^t Q \frac{\partial \mu}{\partial \boldsymbol{\gamma}} + \frac{\partial \mu}{\partial \boldsymbol{\gamma}}^t Q \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \\
&= \tau_0^2 + \left(J_{10} J_{00}^{-1} \frac{\partial \mu}{\partial \boldsymbol{\theta}} - \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \right)^t Q \left(J_{10} J_{00}^{-1} \frac{\partial \mu}{\partial \boldsymbol{\theta}} - \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \right) \\
&= \tau_0^2 + \boldsymbol{\omega}^t Q \boldsymbol{\omega}.
\end{aligned}$$

For the narrow focus parameter, we have that

$$\begin{aligned}
\sqrt{n}(\hat{\mu}_{narr} - \mu_{true}) &= \sqrt{n}(\hat{\mu}(\hat{\boldsymbol{\theta}}_0, \boldsymbol{\gamma}_0) - \mu(\boldsymbol{\theta}_0, \boldsymbol{\gamma}_0 + \boldsymbol{\delta}/\sqrt{n})) \\
&= \sqrt{n}(\hat{\mu}(\hat{\boldsymbol{\theta}}_0, \boldsymbol{\gamma}_0) - \mu(\boldsymbol{\theta}, \boldsymbol{\gamma}_0)) - \sqrt{n}(\mu(\boldsymbol{\theta}, \boldsymbol{\gamma}_0 + \boldsymbol{\delta}/\sqrt{n}) - \mu(\boldsymbol{\theta}, \boldsymbol{\gamma}_0)) \\
&=_D \sqrt{n} \frac{\partial \mu}{\partial \boldsymbol{\theta}} (\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}_0) - \sqrt{n} \frac{\partial \mu}{\partial \boldsymbol{\gamma}} \frac{\boldsymbol{\delta}}{\sqrt{n}} \\
&\to_D \mathcal{N}\left(\boldsymbol{\omega}^t \boldsymbol{\delta}, \tau_0^2 \right).
\end{aligned}$$

■

Under the sequence of models, with $\boldsymbol{\gamma} = \boldsymbol{\delta}/\sqrt{n}$, the proof could be carried out using both the narrow and wide estimates of J and ω, or in fact any model in between. Using the wide estimate gives some robustness, since $\boldsymbol{\gamma}$ does not have to be as close to $\boldsymbol{\gamma}_0$.

Example 1.3.1 (With or without μ) *Take the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with log densiy function*

$$\ell(y; \mu, \sigma) = -\frac{1}{2}\log(2\pi) - \log(\sigma) - \frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2,$$

and score functions

$$S^{(1)} = -\frac{1}{\sigma} + \frac{(y-\mu)^2}{\sigma^3}$$

$$S^{(2)} = -\frac{2}{2\sigma^2}(y-\mu)(-1) = \sum_{i=1}^{n} \frac{y-\mu}{\sigma^2}.$$

Note that we flip the order of μ and σ. Since μ separates the narrow model from the wide, is it more natural to place it last. By the covariance of the score functions we get that

$$J = \frac{1}{\sigma^2}\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \qquad J^{-1} = \sigma^2 \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix}.$$

Assume that we have n observations from a distribution $g_n = \mathcal{N}(\delta/\sqrt{n}, \sigma^2)$, and we wish to estimate the mean. In this case, the narrow model is the $\mathcal{N}(0, \sigma^2)$ distribution, while the wide model is $\mathcal{N}(\mu, \sigma^2)$, where μ is estimated from data. Using the theory above we get that

$$\omega = -1 \qquad Q = \sigma^2 \qquad \tau_0 = 0,$$

which gives that

$$\lim_{n\to\infty} n \cdot mse\,(\hat{\mu}_{narr}) = \omega^2\delta^2 = \delta^2$$

$$\lim_{n\to\infty} n \cdot mse\,(\hat{\mu}_{wide}) = \tau_0^2 + \omega^2 Q = \sigma^2.$$

So under the sequence of distributions $g_n = \mathcal{N}(y; \delta/\sqrt{n}, \sigma)$, the wide estimator is better whenever $\sigma^2 < \delta^2$.

□

In order to state the FIC formulaes, we need to describe the distribution of $\hat{\mu}_S = \mu(\hat{\theta}, \hat{\gamma}_S)$ in the submodels M_S. The submodels all include $\boldsymbol{\theta}$, but each a unique selection of components from $\boldsymbol{\gamma}$. Let $|S|$ denote the number of parameters from $\boldsymbol{\gamma}$ in M_S.

One tool used here are the projection matrices π_S. They are defined as the identity matrix I_q, but where the rows corresponding to the components in $\boldsymbol{\gamma}$ not in γ_S are left out, so π_S is a $|S| \times q$ matrix.

Theorem 1.3.2 *Let $D \sim \mathcal{N}_q(\delta, Q)$ and $\Lambda_0 \sim \mathcal{N}(0, \tau_0^2)$ be two independent random variables. Then*

$$D_n = \hat{\delta} = \sqrt{n}(\hat{\gamma} - \gamma_0) \to_D D \sim \mathcal{N}_q(\delta, Q)$$

For the maximum likelihood estimator of $\hat{\mu}_S$ from submodel S we have that

$$\sqrt{n}(\hat{\mu}_S - \mu_{true}) \to_D \Lambda_S = \Lambda_0 + \boldsymbol{\omega}^t(\delta - G_S D)$$

where $Q_S = (\pi_S^t Q^{-1} \pi_S)^{-1}$ and $G_S = \pi_S Q_S \pi_S^t Q^{-1}$.

For proof see Claeskens and Hjort (2003). The FIC score is n times the sample estimate of the mean squared error of $\hat{\mu}_S$. For the narrow model that is

$$\lim_{n\to\infty} \text{var}\left[\sqrt{n}(\hat{\mu}_{narr} - \mu_{true})\right] = \tau_0^2$$

$$\lim_{n\to\infty} \text{bias}^2\left[\sqrt{n}(\hat{\mu}_{narr} - \mu_{true})\right] = \boldsymbol{\omega}^t \delta \delta^t \boldsymbol{\omega},$$

while for the other models we have

$$\lim_{n\to\infty} \text{var}\left[\sqrt{n}(\hat{\mu}_S - \mu_{true})\right] = \tau_0^2 + \boldsymbol{\omega}^t \pi_S^t Q_S \pi_S \boldsymbol{\omega}$$

$$\lim_{n\to\infty} \text{bias}^2\left[\sqrt{n}(\hat{\mu}_S - \mu_{true})\right] = \boldsymbol{\omega}^t (I_q - G_S)\delta\delta^t (I_q - G_S)\boldsymbol{\omega}.$$

Ways of estimating these variables have already been discussed, except for δ. We know that $E[D_n D_n^t] = \delta\delta^t + Q$, so an estimator for $\delta\delta^t$ is $D_n D_n^t - \hat{Q}$.

1.3.1 The FIC

We are now ready to state the mathematical formulas and framework for the FIC, which were published in Claeskens and Hjort (2003). Let

$$D_n = \sqrt{n}(\hat{\gamma} - \gamma_0)$$

$$\hat{Q} = \hat{J}^{11}$$

$$\hat{\tau}_0^2 = \frac{\partial \mu}{\partial \hat{\boldsymbol{\theta}}_0}^t \hat{J}_{00}^{-1} \frac{\partial \mu}{\partial \hat{\boldsymbol{\theta}}_0}$$

$$\hat{\boldsymbol{\omega}} = \hat{J}_{10}\hat{J}_{00}^{-1} \frac{\partial \mu}{\partial \hat{\boldsymbol{\theta}}_0} - \frac{\partial \mu}{\partial \gamma_0},$$

which are globally defined for every candidate model M_S. For the narrow parameter, $\hat{\mu}_{narr} = \mu(\hat{\boldsymbol{\theta}}_{narr}, \gamma_0)$, we have that

$$\widehat{\text{var}}(\hat{\mu}_{narr}) = \hat{\tau}_0^2$$

$$\widehat{\text{bias}}^2(\hat{\mu}_{narr}) = \hat{\boldsymbol{\omega}}^t(D_n D_n^t - \hat{Q})\hat{\boldsymbol{\omega}}.$$

For the wider models, with $\hat{\mu}_S = \mu(\hat{\boldsymbol{\theta}}_S, \hat{\gamma}_S)$, we have

$$\hat{Q}_S = (\pi_S \hat{Q}^{-1} \pi_S^t)^{-1}$$

$$\hat{G}_S = \pi_S^t \hat{Q}_S \pi_S \hat{Q}^{-1}$$

$$\widehat{\text{var}}(\hat{\mu}_S) = \hat{\tau}_0^2 + (\pi_S \hat{\boldsymbol{\omega}})^t \hat{Q}_S (\pi_S \hat{\boldsymbol{\omega}})$$

$$\widehat{\text{bias}}^2(\hat{\mu}_S) = \hat{\boldsymbol{\omega}}^t (I_q - \hat{G}_S)(D_n D_n^t - \hat{Q})(I_q - \hat{G}_S)^t \hat{\boldsymbol{\omega}}.$$

In either case, the approximate mean squared error, or FIC score is calculated as

$$FIC(S) = \widehat{\text{mse}}(\hat{\mu}_S) = \widehat{\text{var}}(\hat{\mu}_S) + \widehat{\text{bias}}^2(\hat{\mu}_S).$$

The matrices J and ω can be derived using explicit formulae, or could be calculated numerically. In these formulas they are estimated at the narrow model, but they could be replaced with estimates from any submodel M_S.

In Claeskens and Hjort (2008, p 150) a remark is given for cases where the squared bias is negative. The solution is to use a corrected version

$$\text{bias}^2(\hat{\mu}_S)_c = \begin{cases} 0, & \text{bias}^2(\hat{\mu}_S) \leq 0 \\ \text{bias}^2(\hat{\mu}_S), & \text{bias}^2(\hat{\mu}_S) > 0 \end{cases}$$

This bias adjustment is used throughout this entire thesis.

Example 1.3.2 (Lifetime Distributions) *Assume that $y_1, \ldots, y_n$ is a sample of independent random variables, from a probability distribution with density function*

$$f(\theta, \boldsymbol{\gamma}) = \begin{cases} \frac{\gamma_1 \gamma_2 \theta^{\gamma_1}}{\Gamma(1/\gamma_2)} y^{\gamma_1 - 1} \exp\left(-(y\theta)^{\gamma_1 \gamma_2}\right) & y \geq 0 \\ 0 & \textit{otherwise} \end{cases},$$

which is a Weiβull distribution with an added parameter γ_2. The distribution incorporates both the Weiβull and the exponential distribution. The narrow model is in this case the exponential, at $\boldsymbol{\gamma} = \boldsymbol{\gamma}_0 = (1, 1)$. The log likelihood function of f given $y_1, \ldots, y_n$ iid random variables is

$$\ell_n = \sum_{i=1}^{n} \left(\log \gamma_1 + \log \gamma_2 + \gamma_1 \log \theta - \log \Gamma\left(1/\gamma_2\right) + (\gamma_1 - 1) \log y_i - (\theta y_i)^{\gamma_1 \gamma_2}\right).$$

Let the focus parameter μ be the median of the distribuion

$$\mu = F^{-1}\left(\frac{1}{2}\right).$$

It is possible to calculate ω and J analytically, but it takes some effort. Instead we use the numerical methods described in appendix B. The results from $n = 100$ random variables from the distribution with parameters $(\theta, \gamma_1, \gamma_2) = (1/5, 2, 2)$ was

	st.dev	*bias*	*rFIC*	$\hat{\mu}$
Exponential	*2.4149*	*11.9341*	*12.1760*	*2.4149*
Weiβull	*2.6352*	*0.5581*	*2.6937*	*3.3556*
Expanded	*2.6356*	*0.0000*	*2.6356*	*3.4704*

The true mean is 3.4530*, so FIC performed well in this case. A simulation with* 1000 *repetitions of the experiment, tells that the FIC was not far from correct.*

	$\hat{\mu}$	*bias*	$\hat{s}$	*rmse*
Exponential	2.3961	1.1171	0.1065	1.1281
Weiβull	3.3682	0.0366	0.1715	0.1753
Expanded	3.4530	0.0000	0.1812	0.1812

□

1.3.2 The Average-Focussed Information Criterion

In Claeskens and Hjort (2003), the Average-FIC is also presented. Assume that the focus parameter μ varies over some quantity u in the population. This could for example be the observations themself, or covariates in a regression model. Introduce a new loss function

$$L_n(S) = n \int \left(\hat{\mu}_S(u) - \mu_{true}(u)\right)^2 dW_n(u),$$

where W_n is the weight function over the quantity u. The Average-FIC, or limiting loss, is given by

$$AFIC(S) = \max\left\{\hat{I}(S), 0\right\} + \hat{I}I(S),$$

where

$$\hat{I}(S) = Tr\left((I_q - \hat{G}_S)(D_n D_n^t - Q)(I_q - \hat{G}_S)^t \hat{A}\right)$$

$$\hat{I}I(S) = Tr\left(\pi_S^t Q_S \pi_S \hat{A}\right).$$

The matrix $\hat{A}$ is the sample estimate of A, where

$$A = J_{10} J_{00}^{-1} B_{00} J_{00}^{-1} J_{01} - J_{10} J_{00}^{-1} B_{01} - B_{10} J_{00}^{-1} J_{01} + B_{11}$$

$$B = \int_{-\infty}^{\infty} \begin{pmatrix} \frac{\partial \mu}{\partial \theta} \\ \frac{\partial \mu}{\partial \gamma} \end{pmatrix} \begin{pmatrix} \frac{\partial \mu}{\partial \theta} \\ \frac{\partial \mu}{\partial \gamma} \end{pmatrix}^t dW(u) = \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix}.$$

The weight function w_n is ideally a probability density, however that is not necessary. Again, the estimates for $\frac{\partial \mu}{\partial \boldsymbol{\theta}}$, $\frac{\partial \mu}{\partial \gamma}$ and J can with obtained by parameters from any sub model M_S.

1.3.3 About Uncertainty in Model Selection

Assume a model selection situation, with model candidates $\{M_S\}$. The probability of the true model attaining the lowest FIC value might be low. This is discussed in Claeskens and Hjort (2008, Sec. 5.7).

For this thesis, let $\pi_n(S) = P(M_S \text{ is selected})$ be a multinomial distribution with probability distribution

$$\pi_n(\boldsymbol{x}) = \prod_{i=0}^{r} p_n(S_i)^{x_i}.$$

Here r is the number of models to select among, and each model M_S has an index from 0 (narrow) to r (wide). For example if every possible sub model is considered, $r = 2^q$. This probability distribution is non-trivial to compute, since it depends on many variables. This means that given the distribution π_n, the expected value of a focus parameter $\hat{\mu}$ is in fact

$$E[\hat{\mu}_{final}] = \sum_{i=1}^{r} \hat{\mu}(S_i)\pi_n(S_i).$$

In this thesis we are only concerned about its existence, and the possibility of estimating it empirically from simulations. The latter is used to study the bevaviour of FIC and Average-FIC with increasing sample size.

1.4 Kernel Density Estimation

Kernel density estimation was presented in Rosenblatt (1956) and Parzen (1962). Let $y_1, y_2, \ldots, y_n$ be observations from n independent identically distributed random variables, with density function $g(y)$. Both presents the kernel estimate f_n of g as

$$f_n(y) = \int_{-\infty}^{\infty} \frac{1}{h} K\left(\frac{y-t}{h}\right) dG_n(t) = E\left[\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{y-y_i}{h}\right)\right], \tag{1.11}$$

where $K(y)$ is called the kernel function, and h is the bandwidth. Parzen also states that if $\int_{-\infty}^{\infty} K(y)dy = 1$, $h(n)$ is chosen such that

$$\lim_{n\to\infty} h(n) \to 0$$

$$\lim_{n\to\infty} nh(n) \to \infty,$$

the function $K(y)$ is absolutely bounded, and satisfies

$$\lim_{y\to\infty} |yK(y)| = 0,$$

and $\int_{-\infty}^{\infty} |g(y)|dy < \infty$, then $f_n(y)$ is a consistent estimator of $g(y)$ at every continuity point. In this chapter we will also assume that $K(y)$ is symmetric and satisfies

$$\int_{-\infty}^{\infty} tK(y)dt = 0, \qquad \int_{-\infty}^{\infty} t^2K(t)dt = k_2 \neq 0.$$

From Silverman (1986, p. 39) we have that the approximate bias and variance of the kernel estimate at a point z is

$$\text{bias}_h(z) \approx \frac{1}{2}h^2 f''(z)k_2$$

$$\text{var}_h(z) \approx \frac{1}{nh} f(z) \int_{-\infty}^{\infty} K(t)^2 dt.$$

This gives that the mean integrated squared error is

$$MISE(g, \hat{f}_n) = E\left[\int_{-\infty}^{\infty} (g - f_n)^2\right] = \int_{-\infty}^{\infty} \text{bias}_h^2(z) + \text{var}_h(z)dz$$

$$= \frac{1}{4}h^4k_2^2 \int_{-\infty}^{\infty} g''(y)^2 dy + \frac{1}{nh}\int_{-\infty}^{\infty} K(t)^2 dt + o\left(h^4 + \frac{1}{nh}\right). \quad (1.12)$$

1.4.1 Asymptotic Normality of the Kernel Mode

Let $f_n(y)$ be the sequence of functions defined in (1.11), define the sample mode $\hat{y}_{0,K}$ as the point

$$\hat{y}_{0,K} = \arg\max_y \{f_n(y)\}. \quad (1.13)$$

In Parzen (1962) it is proven that if the true mode is unique, and $nh^2 \to \infty$ as $n \to \infty$, then the kernel mode converges in probability to the true mode[2]. The asymptotic normality of the estimated mode is discussed in Parzen (1962), but reviewed in Eddy (1980, Theorem 2.1) under weaker conditions. Let $p \geq 2$ be an integer, and let K be a bounded, absolutely continous function with bounded derivative K'. The next theorem demands that

d1 $B_0 = \int K(t)dt = 1$

d2 $B_i = \int t^i K(t)dt = 0, i = 1, \ldots, p-1$

d3 $B_p, B_{p+1} < \infty$

[2]Convergence with probability one has been shown under stronger conditions in Nadaraya (1965) and Van Ryzin (1969)

d4 $\int [K'(t)]^2 \, dt = V < \infty$

d5 $\int t \, [K'(t)]^2 \, dt < \infty$,

and that $h = h(n)$ is a sequence of positive constants that satisfy

d6 $\lim_{n\to\infty} nh^5 = \infty$

d7 $\lim_{n\to\infty} (nh^{3+2p})^{\frac{1}{2}} = d < \infty$.

Theorem 1.4.1 (Asymptotic normality of the sample mode) *Let K be a function satisfying conditions d1-d5, and let $h = h(n)$ be a sequence of positive constants satisfying d6-d7. If the density f is bounded, has an absolutely bounded $(p+1)$st derivative and satisfies*

$$\sup_t |g^{(i)}(t)| < \infty$$

then

$$(nh^3)^{\frac{1}{2}} (\hat{y}_{0,K} - y_0) \to_D \mathcal{N}\left((-1)^p \cdot \frac{d}{p!} \cdot \frac{g^{(p+1)}(\theta)}{g^{(2)}(\theta)} \cdot B_p, \frac{g(y_0)}{[g''(y_0)]^2} V \right)$$

where $V = \int_{-\infty}^{\infty} [K'(t)]^2 dt$.

1.4.2 Bandwidth Selection for Density Estimation

For the Density

In Parzen (1962, Lemma 4A), it is shown that minimizing (1.12) is equivalent to choosing h_{opt} to be

$$h_{opt} = k_2^{-2/5} \left(\int_{-\infty}^{\infty} g''(y)^2 dy \right)^{\frac{1}{5}} \left(\int_{-\infty}^{\infty} K(t)^2 dt \right)^{-\frac{1}{5}} n^{-\frac{1}{5}},$$

which is non-trivial to compute, since h_{opt} depends on the second derivative of the unknown density. Silverman (1986, p. 45) suggests using Gaussian kernel, and insert $g = \mathcal{N}(\mu, \sigma)$. In that case it can be shown that

$$h = 1.059 \sigma n^{-\frac{1}{5}}$$

is the optimal h for minimizing the MISE. However, this might oversmooth in cases of multimodality, if $(\int g''(y) dy)^{1/5}$ is large relative to σ. A discussion about this problem is found in Silverman (1986, p. 46), and his solution is to use the same rule-of-thumb, but adjusted for larger values of $(\int g''(y) dy)^{1/5}$

induced by multimodality in normal mixtures. His modified rule-of-thumb bandwidth is

$$h_{dens} = 0.9An^{-\frac{1}{5}}, \qquad A = \min\left\{\sigma, \frac{Q_3 - Q_1}{1.34}\right\}, \tag{1.14}$$

where $Q_3 - Q_1$ is the interquartile range. This will be used for density estimation throughout this entire thesis.

For The Third Derivative

In this section, we wish to establish a rule-of-thumb for third derivative estimation. We have that the bias of the triple derivative estimator is

$$\text{bias}\left(\hat{f}_n^{(3)}(z)\right) = \frac{1}{2}h^2 g^{(5)}(z)k_2,$$

which follows directly from the bias term for $\hat{f}_n$. The variance is

$$\text{var}(\hat{f}_n^{(3)}(y_0)) = \frac{1}{n}\int_{-\infty}^{\infty}\left[\frac{1}{h^4}K^{(3)}\left(\frac{z-y}{h}\right)\right]^2 g(y)dy - \left(\frac{1}{2}h^2 g^{(5)}(z)k_2\right)^2.$$

Changing the variable in the integral to $y = z - ht$ gives

$$\frac{1}{n}\int_{-\infty}^{\infty}\frac{1}{h^7}\left(K^{(3)}(t)\right)^2 g(z-ht)dt - \left(\frac{1}{2}h^2 g^{(5)}(z)k_2\right)^2.$$

Assume that h is small and n is large. Using a Taylor series expansion of g around z we get that

$$\begin{aligned}\text{var}(\hat{f}_n^{(3)}(z)) &= \frac{1}{nh^7}\int_{-\infty}^{\infty}\left[g(z) - htg'(z) + o(z-ht)^2\right]\frac{1}{h^7}\left(K^{(3)}(t)\right)^2 dt + o\left(\frac{1}{nh^7}\right)\\ &\approx \frac{1}{nh^7}g(z)\int_{-\infty}^{\infty}\left(K^{(3)}(t)\right)^2 dt.\end{aligned}$$

By putting the bias and variance together, and integrating with respect to z, we get that

$$\text{mise}(\hat{f}_n^{(3)}) \approx \frac{1}{nh^7}\int_{-\infty}^{\infty}\left(K^{(3)}(t)\right)^2 dt + \frac{1}{4}h^4 k_2^2\int_{-\infty}^{\infty}\left(g^{(5)}(z)\right)^2 dz.$$

Differentiating with respect to h, gives that the optimal h must satisfy the equation

$$\lim_{n\to\infty} nh^{11} = 7\int_{-\infty}^{\infty}\left(K^{(3)}(t)\right)^2 dt\left[k_2^2\int_{-\infty}^{\infty}\left(g^{(5)}(z)\right)^2 dz\right]^{-1}.$$

This is very hard to compute empirically for a general distribution. However, one can establish a rule-of-thumb similar to that of Silverman, by choosing K as

the standard normal distribution, and substitute g with a normal distribution $\mathcal{N}(0,\sigma)$. First we have that

$$\int_{-\infty}^{\infty}\left(K^{(3)}(t)\right)^2 dz = \int_{-\infty}^{\infty}\left(\phi(z)(z^3-3z)\right)^2 dz \approx 0.5289$$

$$\int_{-\infty}^{\infty}\left(g^{(5)}(z)\right)^2 dz = \frac{1}{\sigma^{22}}\int_{-\infty}^{\infty}\left(\phi(z)(-15\sigma^4 x + 10\sigma^2 z^3 - z^5)\right)^2 dz \approx \frac{8.3305}{\sigma^{11}},$$

which means that the optimal h for the third derivative must satisfy

$$\lim_{n\to\infty} nh^{11} = 7\cdot 0.5289\cdot\left(\frac{8.3305}{\sigma^{11}}\right)^{-1} \approx 0.4444\sigma^{11}.$$

This gives that our rule of thumb bandwidth is

$$h = 0.9289\sigma n^{-\frac{1}{11}}.$$

1.4.3 Bandwidth Selection for the Mode

Eddy (1980, Eq 3.1) shows that the mean squared error of the mode estimator is

$$E[(\hat{y}_{0,K}-y_0)^2] = \left[\frac{h^p\cdot B_p\cdot f^{(p+1)}(y_0)}{p!g^{(2)}(y_0)}\right]^2 + \frac{g(y_0)V}{nh^3[g^{(2)}(y_0)]^2}.$$

Differentiating this with respect to h gives

$$p\cdot h^{2p-1}\left[\frac{B_p\cdot g^{(p+1)}(y_0)}{p!\cdot g^{(2)}(y_0)}\right]^2 - \frac{g(y_0)\cdot V}{3\cdot n\cdot h^4[g^{(2)}(y_0)]^2} = 0,$$

so the optimal h must statisfy

$$\lim_{n\to\infty} nh^{2p+3} = \left[\frac{p!\cdot g^{(2)}(y_0)}{B_p\cdot g^{(p+1)}(y_0)}\right]^2\cdot\frac{g(y_0)\cdot V}{3p\cdot[g^{(2)}(y_0)]^2} = \left[\frac{p!}{B_p\cdot g^{(p+1)}(y_0)}\right]^2\cdot\frac{g(y_0)\cdot V}{3p}.$$

Assume that the kernel K is the standard normal distribution $\phi(t)$. Then $B_1=0$ and $B_2=1$, so $p=2$. In this case, the optimal h must satisfy

$$\lim_{n\to\infty} nh^7 = \frac{2\cdot g(y_0)\cdot\left(\sqrt{2\pi}\right)^{-1}}{3\left[g^{(3)}(y_0)\right]^2}. \tag{1.15}$$

To estimate the bandwidth, an initial $\hat{y}_{0,first}$ har to be estimated. For simplicity, both $\hat{y}_{0,first}$ and $g(\hat{y}_{0,first})$ are estimated with Silvermans rule of thumb. After that, $g'''(\hat{y}_{0,first})$ is estimated with the rule of thumb for the third derivative, and numerical differentiation. The numerical calculation of the third derivative is described in appendix B.1.3.

Note About Symmetric Distributions

Using (1.15) with g as the normal distribution is not possible, since $g'''(y_0)$ would is zero, and result in the optimal bandwidth being infinite. This is obviously not feasible as a general rule, but for the normal distribution it makes more sense.

Eddy (1982) showed that if g is symmetric about the mode, and K is symmetric, then there is no asymptotic bias effect, and the mean squared error is

$$\text{mse}(\hat{y}_{0,K}) = \frac{g(y_0)V}{nh^3[g^{(2)}(y_0)]^2},$$

which is small for a very large h. The optimal h is ∞, but there are limitations of what h one can choose to make the asymptotic results valid. Assume a kernel estimate with Gaussian kernel

$$\hat{f}_n = \frac{1}{nh}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{y-y_i}{h}\right)^2\right).$$

Differentiating with respect to y gives

$$\frac{1}{nh}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{y-y_i}{h}\right)^2\right)\left(-\frac{y-y_i}{h}\right) = 0$$

$$\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\left(\frac{y-y_i}{h}\right)^2\right)\left(\frac{y-y_i}{h}\right) = 0.$$

Since $1/h^2$ goes to zero faster than $1/h$, and the first term converges to one, we get that the mode converges to the solution of

$$\sum_{i=1}^{n}\left(\frac{y-y_i}{h}\right) = 0.$$

This tells that the kernel mode converges to the sample mean for large h. This gives meaning to (1.15) when $g^{(3)}(y_0) = 0$. For the normal distribution, it will result in the UMVU estimator for the mode, namely the mean.

Potential Problems with Multimodality

Another problem with the bandwidth presented, is that it does not detect multimodality. Distributions can be have a low third derivative at the mode compared to σ, which could lead to oversmoothing.

Chapter 2

A Mode Hunter's Scheme

Barron and Sheu (1991) discusses the concept of log expanding existing probability distributions, in order to make them more flexible. Examples of expansion functions are polynomials, splines or trigonometric series. The general form in this thesis is

$$f_m(y;\boldsymbol{a}) = f_0 \cdot \exp\left(\sum_{k=1}^{m} a_k \psi_k(F_0(y))\right) \frac{1}{k_m(\boldsymbol{a})}, \quad k_m(a) = \int_0^1 \exp\left(\sum_{k=1}^{m} a_k \psi_k(y)\right) dy,$$

where $\{\psi_k\}$ is a family of orthonormal functions, and f_0 is a probability distribution with cumulative distribution F_0. These families of density functions have good properties in terms of convergence in the Kullback-Leibler divergence. The objective for this chapter is to develop a machinery that can compete with the kernel estimate in mode hunting and density estimation.

2.1 An Orthonormal Family

Parts of this section is not directly linked to the main discussion in this thesis, but to build a foundation for further functional analysis, and to explain some of the strength found in the log expanded distribution families. For the rest of this thesis, we define the family $\{\psi_k\}$ as

$$\psi_k(u) = \begin{cases} 1, & k = 0 \\ \sqrt{2}\cos(k\pi u), & k \geq 1 \end{cases}.$$

Note that even though we use the cosine series in this thesis, any bounded orthonormal function family will work. The following lemma shows the orthonormality of the cosine series.

Lemma 2.1.1 *The functions $\psi_j(u) = \sqrt{2}\cos(j\pi u)$, $\psi_0 = 1$ for $j = 0, 1, \ldots$, are orthonormal with respect to the L_2-norm on $[0, 1]$. That is*

$$\int_0^1 \psi_j(u)\psi_k(u)du = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases}.$$

Proof: For $j = k$ and $j, k \geq 1$, we have that

$$\int_0^1 \psi_j(u)\psi_k(u)du = 2\int_0^1 \cos^2(j\pi u)du = 2\int_0^1 \frac{1}{2}\left(\cos(2\pi ju) + 1\right)du$$

$$= \left[\frac{\sin(2\pi ju)}{2\pi j} + u\right]_0^1 = 1.$$

For $j \neq k$ and $j, k \geq 1$ we have that

$$\int_0^1 \psi_j(u)\psi_k(u)du = 2\int_0^1 \cos(j\pi u)\cos(k\pi u)du$$

$$= 2\int_0^1 \cos((j-k)\pi u) + \cos((j+k)\pi u)du$$

$$= \left[\frac{\sin((j-k)\pi u)}{(j-k)\pi} + \frac{\sin((j+k)\pi u)}{(j+k)\pi}\right]_0^1 = 0,$$

and for ψ_0 and ψ_j and $j \geq 0$, we have

$$\int_0^1 1 \cdot \cos(j\pi u)du = \left[\frac{\sin(j\pi u)}{j\pi}\right]_0^1 = 0.$$

The innerproduct of ψ_0 with 1 as well.

■

One peculiar observation is that it seems that for continous random variables, $\text{cov}[\psi_j(y), \psi_k(y)] \approx 0$ for $k > 0$, while $\text{var}[\psi_k(y)] \approx 1$. This seems to hold for any normal random variable with $\sigma > 0.5$, and some exponential, gamma and log normal random variables. However, the conditions are unclear and beyond the scope of this thesis.

Similar observations has been done with discrete random variables as well. For some binomial and poisson random variables the variance and covariance of the odd numbered ψ_k are 2, while for even numbered they are zero.

Note that these are just insinuations and needs theoretical verification, but a theoretical result backing the hypotheses would be useful for future work on estimators containing this family of functions.

2.1.1 L_2 Norm Convergence for $\{\psi\}$

Assume now that we want to estimate a function $g(u) \in L_2[0,1]$ with a sequence of ψ-functions. In the L_2 norm, we need to coefficients $\{a_j\}$ that minimizes

$$\int_0^1 \left(\sum_{k=0}^{m} a_k \sqrt{2} \cos(k\pi u) - g(u) \right)^2 du.$$

The derivatives with respect to each a_j must satisfy the equation

$$\int_0^1 2 \left(\sum_{k=0}^{m} a_k \sqrt{2} \cos(k\pi u) - g(u) \right) \sqrt{2} \cos(j\pi u) = 0,$$

which implies that

$$\int_0^1 \left(\sum_{k=0}^{m} a_k \sqrt{2} \cos(k\pi u) \right) \sqrt{2} \cos(j\pi u) du = \int_0^1 g(u) \sqrt{2} \cos(j\pi u) du.$$

We know from earlier that ψ_j is an orthonormal family, so the terms in the left integral are zero, except from when $k = j$. This gives that the coefficient estimates are

$$a_0 = \int_0^1 g(u) du \tag{2.1}$$

$$a_j = \int_0^1 g(u) \sqrt{2} \cos(j\pi u) du. \tag{2.2}$$

So how close does $\{\psi_j\}$ get to g? We have that

$$\begin{aligned}
&\int_0^1 \left(\sum_{k=0}^{m} \sqrt{2} a_k \cos(k\pi u) - g(u) \right)^2 du \\
&= \int_0^1 \left(\sum_{k=0}^{m} \sqrt{2} a_k \cos(k\pi u) \right)^2 - 2 \sum_{k=0}^{n} \sqrt{2} a_k \cos(k\pi u) g(u) + g(u)^2 du \\
&= \sum_{k=0}^{n} a_k^2 - 2 \sum_{k=0}^{m} a_k^2 + \int_0^1 g(u)^2 du = \int_0^1 g(u)^2 du - \sum_{k=0}^{n} a_k^2 \\
&= \int_0^1 g(u)^2 du - \sum_{k=0}^{m} \left(\int_0^1 g(u) \sqrt{2} \cos(k\pi u) du \right)^2.
\end{aligned}$$

Since the L_2 inner product space is a Hilbert space, we have from Bessel's inequality (see Teschl (2011, p. 36)) that

$$\sum_{k=0}^{m} \left(\int_0^1 g(u) \sqrt{2} \cos(k\pi u) du \right)^2 \leq \int_0^1 g(u)^2 du,$$

but since the left side is a monotone growing sequence of numbers, the sum must converge as $m \to \infty$. For the error, we have that

$$\int_0^1 \left(\sum_{k=0}^{m} \sqrt{2} a_k \cos(k\pi u) - g(u) \right)^2 du = \sum_{k=m+1}^{\infty} a_k^2.$$

Assume $k > m$. We have that

$$\int_0^1 g(u)\sqrt{2}\cos(k\pi u)du = \left[\sqrt{2} g'(u) \frac{\sin(k\pi u)}{k\pi} \right]_0^1 - \int_0^1 g'(u)\sqrt{2}\frac{\sin(k\pi u)}{k\pi}du.$$

Since the first term is zero, the error is

$$\sum_{k=m+1}^{\infty} \left(\int_0^1 g'(u)\sqrt{2}\frac{\sin(k\pi u)}{k\pi}du \right)^2 \leq \sum_{k=m+1}^{\infty} \int_0^1 \left[g'(u)\right]^2 \frac{(\sqrt{2}\sin(k\pi u))^2}{k^2\pi^2}du,$$

which gives that

$$E \leq \frac{M}{\pi^2} \sum_{i=m+1}^{\infty} \frac{1}{k^2} = \frac{M}{\pi^2}\eta(m+1) = \mathcal{O}(m^{-1}),$$

where $M = \max_{0\leq u\leq 1}[g'(u)^2]$ and η is the trigamma function, or second derivative of $\log\Gamma(u)$. The error term can be generalized to a function on any interval $[a, b]$ by transforming to the $[0, 1]$-interval, do the estimation, and then transform back. In that case, with integration by substitution, the error term is

$$E \leq (b-a)\frac{M}{\pi^2}\eta(m+1) = \mathcal{O}(m^{-1}).$$

2.1.2 L_2 bound on the Kullback-Leibler Divergence

Assume now that the density we wish to estimate is on the form

$$g(y) = \begin{cases} \exp\left[c(y;\theta)\right] & 0 \leq y \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

with a model f on the form

$$f_m(y;\boldsymbol{a}) = \exp\left(\sum_{j=0}^{m} a_j \psi(y) \right) \frac{1}{k_m(\boldsymbol{a})}$$

Then the KL divergence is

$$\begin{aligned} D_{KL}(g \parallel f_m(y;\boldsymbol{a})) &= \int_0^1 g(y) \frac{\log g(y)}{\log f_m(y;\boldsymbol{a})} dy \\ &= \int_0^1 g(y) \left[c(y;\theta) - \sum_{j=1}^{m} a_j \psi_j(y) + \log k_m(\boldsymbol{a}) \right] dy. \end{aligned}$$

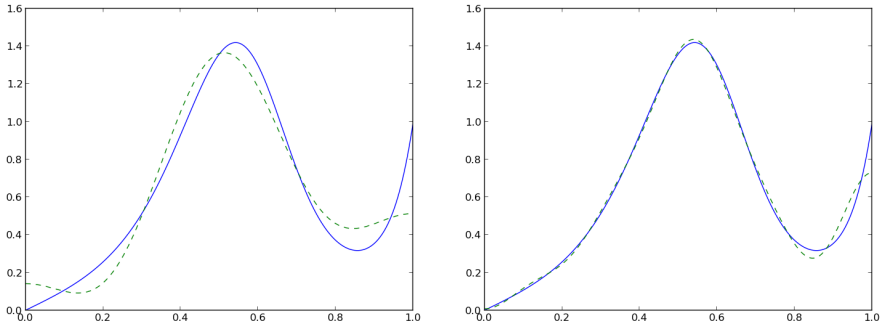


Figure 2.1: Two examples of function estimation with coefficient estimates (2.1) and (2.2). The function used in both plots is $f(x) = x\exp(\sin(2\pi x^2))$, which is plotted with a solid line. The estimates with $n = 5$ (left) and $n = 10$ (right) are plotted with dashed lines.

Since the integral is positive and bounded, the integrand must be square integrable, which means that

$$[D_{KL}(g \parallel f_m(y;\boldsymbol{a}))]^2 \leq \max_{0\leq y\leq 1}\{g(y)^2\}\int_0^1 \left(c(y;\theta) - \sum_{j=1}^{m} a_j\psi_k(y) + \log k_m(\boldsymbol{a})\right)^2 dy$$

$$\leq \left[\max_{0\leq y\leq 1}\{g(y)^2\}\right]\left[\frac{1}{\pi^2}\max_{0\leq y\leq 1}\left[\left(c'(y;\theta)\right)^2\right]\eta(m+1)^2\right].$$

This gives that the Kullback-Leibler divergence over the interval $[0,1]$ is bounded by

$$D_{KL}(g \parallel f_m(y;\boldsymbol{a})) \leq \frac{M}{\pi}\eta(m+1),$$

where $M = \max_{a\leq y\leq b, a\leq z\leq b}\{g(y)\,|c'(z;\theta)|\}$. Again, for any finite interval $[a,b]$, where the data are transformed to $[0,1]$, the bound is

$$D_{KL}(g \parallel f_m(y;\boldsymbol{a})) \leq (b-a)\frac{M}{\pi}\eta(m+1).$$

2.2 Class Definition and Likelihood

In this thesis we will look at one specific class of distributions. It is a trigonometric log expanded normal distribution, with density function

$$f_m(y;\xi,\sigma,\boldsymbol{a}) = \phi\left(\frac{y-\xi}{\sigma}\right)\frac{1}{\sigma}\exp\left(\sum_{j=1}^{m} a_j\psi_j\left(\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right)\frac{1}{k_m(\boldsymbol{a})}, \qquad (2.3)$$

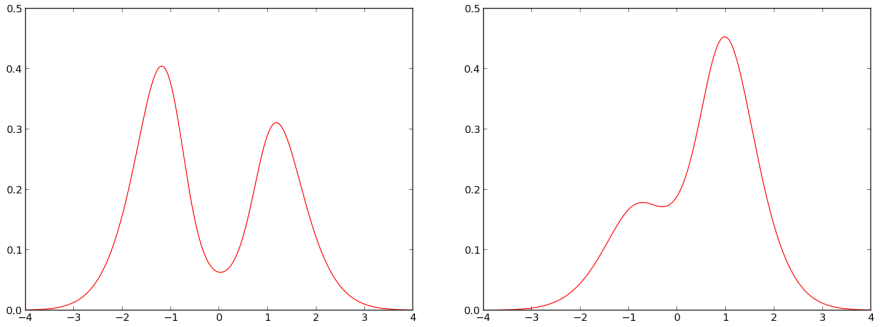


Figure 2.2: Two examples from class (2.3) for $m = 2$. The parameters used are $\xi = 0$ and $\sigma = 1$ for both figures, and $(a_1, a_2) = (0.1, 0.99)$ to the left and $(a_1, a_2) = (-0.4, 0.4)$ to the right.

where $\psi_j(u) = \sqrt{2}\cos(j\pi u)$. This family is also discussed in Claeskens and Hjort (2004). To get an impression of the flexibility of this class, see figure 2.2. For the normalizing constant $k_m(\boldsymbol{a})$, we have

$$k_m(a) = \int_{-\infty}^{\infty} \phi\left(\frac{y-\xi}{\sigma}\right)\frac{1}{\sigma}\exp\left(\sum_{j=1}^{m} a_j\psi_j\left(\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right)dy.$$

Substituting $u = \Phi\left(\frac{y-\xi}{\sigma}\right)$ gives

$$k_m(a) = \int_{\Phi(-\infty)}^{\Phi(\infty)} \exp\left(\sum_{j=1}^{m} a_j\psi_j(u)\right)du = \int_0^1 \exp\left(\sum_{j=1}^{m} a_j\psi_j(u)\right)du,$$

which can be computed using numerical quadrature.

2.2.1 Information matrix for when $\boldsymbol{a} = \mathbf{0}$

In order to calculate FIC, we need the information matrix J estimated at the narrow model. In this case we define the narrow model as any the model with $\boldsymbol{a} = \mathbf{0}$, eg. the normal distribution. The log density function of (2.3) is

$$\begin{aligned}\ell(\boldsymbol{y}; \xi, \sigma, \boldsymbol{a}) &= \log\left[\phi(\varepsilon)\frac{1}{\sigma}\right] + \sum_{j=1}^{m} a_j\psi_j(\Phi(\varepsilon)) - \log(k_m(\boldsymbol{a})) \\ &= -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2}\varepsilon^2 + \sum_{j=1}^{m} a_j\psi_j(\Phi(\varepsilon)) - \log(k_m(\boldsymbol{a})),\end{aligned}$$

where $\varepsilon = \frac{y-\xi}{\sigma}$. The partial derivatives of the log density function is

$$\frac{\partial \ell}{\partial \xi} = \frac{\varepsilon}{\sigma} + \frac{\partial}{\partial \xi} \sum_{j=1}^{m} a_j \psi_j(\Phi(\varepsilon)) \tag{2.4}$$

$$\frac{\partial \ell}{\partial \sigma} = -\frac{1}{\sigma} - \varepsilon(-1)\frac{y-\xi}{\sigma^2} + \frac{\partial}{\partial \sigma} \sum_{j=1}^{m} a_j \psi_j(\Phi(\varepsilon)) \tag{2.5}$$

$$\frac{\partial \ell}{\partial a_j} = \psi_j(\Phi(\varepsilon)) - \frac{\partial}{\partial a_j} \log \left[\int_0^1 \exp\left(\sum_{j=1}^{m} a_j \psi_j(u) \right) du \right]. \tag{2.6}$$

Since this is calculated at the narrow model, $\boldsymbol{a} = \mathbf{0}$, so the sums in (2.4) and (2.5) are 0. For (2.6) we have that

$$\begin{aligned}
&\frac{\partial}{\partial a_j} \log \left[\int_0^1 \exp\left(\sum_{j=1}^{m} a_j \psi_j(u) \right) du \right] = \frac{\partial}{\partial a_j} \int_0^1 \exp\left(\sum_{j=1}^{m} a_j \psi_j(u) \right) du / k_m(\boldsymbol{a}) \\
&= \int_0^1 \frac{\partial}{\partial a_j} \exp\left(\sum_{j=1}^{m} a_j \psi_j(u) \right) du / k_m(\boldsymbol{a}) = \int_0^1 \psi_j(u) \exp\left(\sum_{j=1}^{m} a_j \psi_j(u) \right) du / k_m(\boldsymbol{a}) \\
&= \frac{1}{k_m(\boldsymbol{a})} \int_0^1 \psi_j(u) du = 0.
\end{aligned}$$

This gives that evaluated at $\boldsymbol{a} = \mathbf{0}$, we have that

$$S = \begin{pmatrix} \frac{\varepsilon}{\sigma} \\ \frac{1}{\sigma}\left(\varepsilon^2 - 1\right) \\ \psi_1(\Phi(\varepsilon)) \\ \vdots \\ \psi_m(\Phi(\varepsilon)) \end{pmatrix}.$$

In addition we need the covariance matrix of S. We have that

$$E[\psi_j(\Phi(\varepsilon))] = \int_{-\infty}^{\infty} \psi_j(\Phi(\varepsilon))\phi(\varepsilon)d\varepsilon = \int_0^1 \psi_j(u)du = 0$$

$$E[\psi_j(\Phi(\varepsilon))^2] = \int_{-\infty}^{\infty} \psi_j(\Phi(\varepsilon))^2\phi(\varepsilon)dy = \int_0^1 \psi_j(u)^2 du = 1$$

$$E[\psi_j(\Phi(\varepsilon))\psi_k(\Phi(\varepsilon))] = \int_{-\infty}^{\infty} \psi_j(\Phi(\varepsilon))\psi_k(\Phi(\varepsilon))\phi(\varepsilon)dy = \int_0^1 \psi_j(u)\psi_k(u)du = 0,$$

which implies that $\text{var}(\psi_k(\Phi(\varepsilon))) = 1$ and $\text{cov}(\psi_j(\Phi(\varepsilon)), \psi_k(\Phi(\varepsilon))) = 0$. We also have that $\text{var}\left(\frac{\varepsilon}{\sigma}\right) = \frac{1}{\sigma^2}$, $\text{var}\left(\frac{1}{\sigma}(\varepsilon^2 - 1)\right) = \frac{1}{\sigma^2}\text{var}(\varepsilon^2) = \frac{2}{\sigma^2}$ and

$$\operatorname{cov}\left[\frac{\varepsilon}{\sigma}, \frac{1}{\sigma}(\varepsilon^2 - 1)\right] \propto \operatorname{cov}[\varepsilon, \varepsilon^2] = 0$$

$$\operatorname{cov}\left[\frac{\varepsilon}{\sigma}, \psi_j(\Phi(\varepsilon))\right] = \frac{1}{\sigma}\operatorname{cov}\left[\varepsilon, \psi_j(\Phi(\varepsilon))\right] = \frac{c_j}{\sigma}$$

$$\operatorname{cov}\left[\frac{1}{\sigma}\left(\varepsilon^2 - 1\right), \psi_j(\Phi(\varepsilon))\right] = \frac{1}{\sigma}\operatorname{cov}\left[\varepsilon^2, \psi_j(\Phi(\varepsilon))\right] = \frac{d_j}{\sigma}.$$

The two former are computed by the integrals

$$c_j = E[\varepsilon\psi_j(\Phi(\varepsilon))] = \int_{-\infty}^{\infty} \varepsilon\psi_j(\Phi(\varepsilon))\phi(\varepsilon)d\varepsilon$$

$$d_j = E[\varepsilon^2\psi_j(\Phi(\varepsilon))] = \int_{-\infty}^{\infty} \varepsilon^2\psi_j(\Phi(\varepsilon))\phi(\varepsilon)d\varepsilon,$$

which are computed numerically. This gives that the narrow evaluated estimate for J is

$$J = \begin{pmatrix} \frac{1}{\sigma^2} & 0 & \frac{c_1}{\sigma} & \frac{c_2}{\sigma} & \dots & \frac{c_m}{\sigma} \\ 0 & \frac{2}{\sigma^2} & \frac{d_1}{\sigma} & \frac{d_2}{\sigma} & \dots & \frac{d_m}{\sigma} \\ \frac{c_1}{\sigma} & \frac{d_1}{\sigma} & 1 & 0 & \dots & 0 \\ \frac{c_2}{\sigma} & \frac{d_2}{\sigma} & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & \ddots & \vdots \\ \frac{c_m}{\sigma} & \frac{d_m}{\sigma} & 0 & \dots & \dots & 1 \end{pmatrix}. \tag{2.7}$$

It can be shown that $c_i = 0$ for even i, and that $d_i = 0$ for odd i.

2.2.2 Information Matrix for the Full Model

The wide model is in this case the full $f_m(y; \xi, \sigma, \boldsymbol{a})$ where $a \in \mathbb{R}^m$. Deriving the score function and information matrix for the wide model analytically takes some effort, so we will stick to numerical calculations. The estimate for J_{wide} is then the empirical hessian

$$J(\hat{\boldsymbol{\zeta}}) = -\frac{1}{n}\sum_{i=1}^{n} \frac{\partial^2 \ell(y_i; \boldsymbol{\zeta})}{\partial\boldsymbol{\zeta}\partial\boldsymbol{\zeta}^t}.$$

writing $\boldsymbol{\zeta} = (\xi, \sigma, \boldsymbol{a})$. The double differentiation is done with the algorithm in appendix B.

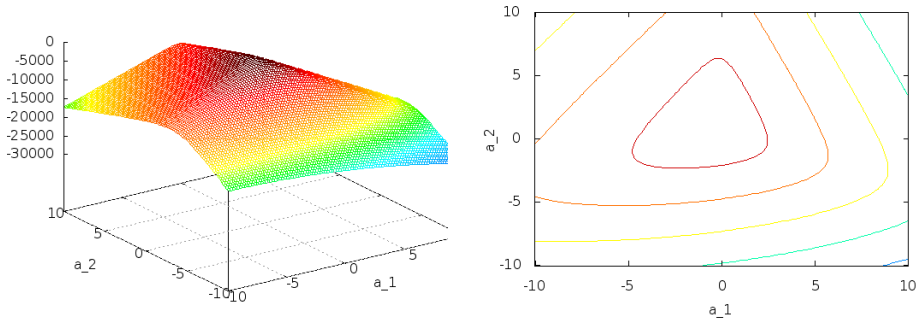


Figure 2.3: Mesh and contour plot of the log likelihood function of (2.3), with $m = 2$ and $n = 1000$ random variables from (3.1) with parameters $\boldsymbol{\mu} = (-1, 2)$, $\boldsymbol{\tau} = (1.1, 0.7)$ and $\boldsymbol{p} = (0.4, 0.6)$. The plot is computed around the maximum likelihood estimates of (ξ, σ, a_1, a_2), but a_1 and a_2 are drawn from -10 to 10.

2.2.3 Investigating the Likelihood Function

Some attempts at fitting data reveals that non linear maximizers in Sci-Py[1], which are used here, struggle to converge properly. In some cases, different starting values may give different results.

For the investigation I've used a dataset of $n = 1000$ from (3.1), with parameters from model 2 as in section (3.4).

The plots of a_1 versus a_2 in figure 2.3 shows good behaviour. Between ξ and a_1 on the other hand, a zig zag shaped ridge appears on figure 2.4, and the ridge has a very low gradient in certain directions.

Further investigation show that flat areas appear on the plot of a_7 against ξ in figure (2.5), and a_7 against σ in figure 2.6. In figure 2.5, there are som irregularities around the maximum as well. One way to encounter the problem is to

I Start with the narrow model, with mean and standard deviation as ML estimates. Start with 0 as the starting value for a_0 and re-estimate the parameters. Repeat this for every a until the correct number of a's are estimated.

II The computations has ended up at some point $\boldsymbol{\theta}_{mid}$, which is probably along a ridge. Create an augmented log likelihood function

$$\ell_{n,aug} = [\ell_n(\boldsymbol{\theta}) - \ell_n(\boldsymbol{\theta}_{mid})] \cdot 10^d$$

[1]Scientific Python, included in the Scitools package discussed in Langtangen (2010).

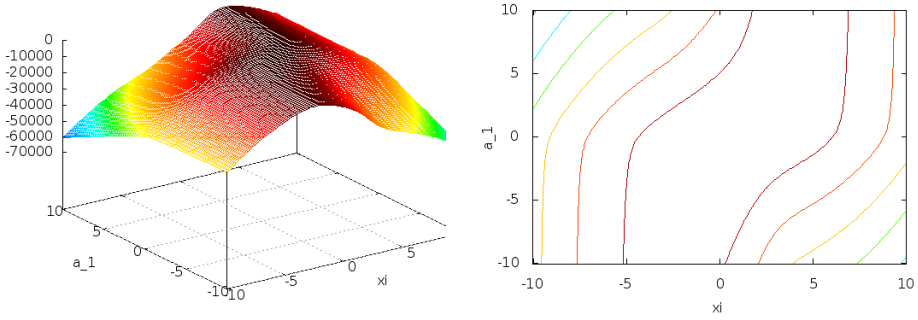


Figure 2.4: Mesh and contour plot of the log likelihood function of (2.3), with $m = 2$ and $n = 1000$ random variables from (3.1) with parameters $\boldsymbol{\mu} = (-1, 2)$, $\boldsymbol{\tau} = (1.1, 0.7)$ and $\boldsymbol{p} = (0.4, 0.6)$. The plot is computed around the maximum likelihood estimates of (ξ, σ, a_1, a_2), but ξ and a_1 are drawn from -10 to 10.

for a suitable value of d, and optimize again.

Instead of just a zero for the next a_j, one could use more thorough approaches. One is to make a grid for the next a, and pick the point along the grid with highest log likelihood value, and use that as the starting point.

Another way could be to make a 2d grid of fitted models with different starting values for ξ and σ, and choose the one with highest likelihood. This is however very demanding in terms of processing power.

2.3 Computing Omega

In terms of FIC estimation, a vector ω has to be calculated, defined as

$$\omega = J_{10} J_{00}^{-1} \frac{\partial \mu}{\partial \boldsymbol{\theta}} - \frac{\partial \mu}{\partial \boldsymbol{a}}.$$

As the information matrix J was covered earlier, this section covers the partial derivatives of the focus parameters. These can be calculated either analytically or numerically, or both to check the calculations. For the numerical approach one can calculate

$$\frac{\partial \mu}{\partial \theta_i} \approx \frac{\mu(\boldsymbol{\theta} + he_i) - \mu(\boldsymbol{\theta} - he_i)}{2h}$$

for every $\theta_i \in \boldsymbol{\theta}$, where e_i is the appropriate unity vector in $\mathbb{R}^p$. Expressions for γ are completely analog.

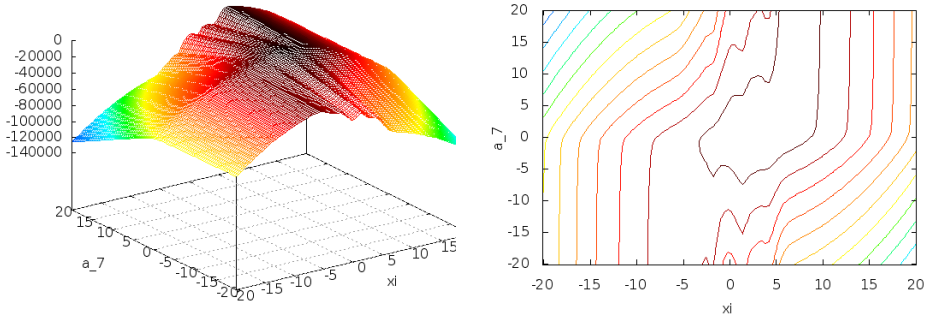


Figure 2.5: Mesh and contour plot of the log likelihood function of (2.3), with $m = 7$ and $n = 1000$ random variables from (3.1) with parameters $\boldsymbol{\mu} = (-1, 2)$, $\boldsymbol{\tau} = (1.1, 0.7)$ and $\boldsymbol{p} = (0.4, 0.6)$. The plot is computed around the maximum likelihood estimates of $(\xi, \sigma, \boldsymbol{a})$, but ξ and a_7 are drawn from -10 to 10.

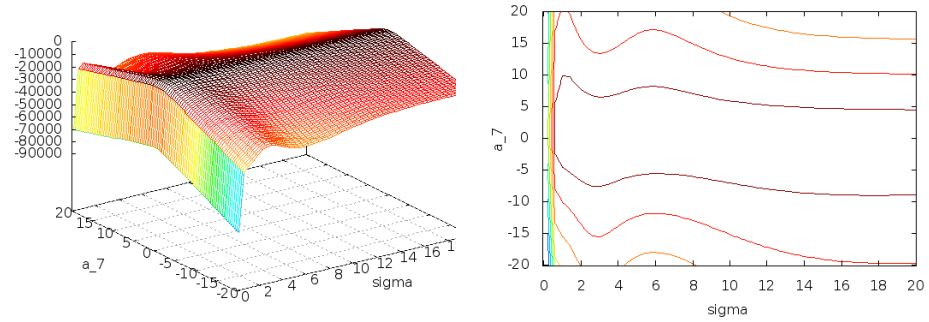


Figure 2.6: Mesh and contour plot of the log likelihood function of (2.3), with $m = 7$ and $n = 1000$ random variables from (3.1) with parameters $\boldsymbol{\mu} = (-1, 2)$, $\boldsymbol{\tau} = (1.1, 0.7)$ and $\boldsymbol{p} = (0.4, 0.6)$. The plot is computed around the maximum likelihood estimates of $(\xi, \sigma, \boldsymbol{a})$, but σ and a_7 are drawn from 0 to 20, and -10 to 10 respectively.

The ω can be computed with narrow or wide parameters. In this case the narrow ω would be when $\boldsymbol{a} = 0$ and (ξ, σ) are estimates by the sample mean and variance. The wide ω is calculated at the maximum likelihood estimate of the wide model. Both narrow and wide estimates are considered in this thesis for comparison.

2.3.1 Omega for the Mode

The first focus parameter is about bumps, which are defined as the solution y_0 to $f'_m(y) = 0$. A quick rewrite of (2.3) gives

$$f_m(y; \xi, \sigma, \boldsymbol{a}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{y-\xi}{\sigma}\right)^2 + \sum_{j=1}^{m}\left[a_j\sqrt{2}\cos\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right]\right)\frac{1}{k_m(\boldsymbol{a})}. \tag{2.8}$$

We have that

$$f'_m(y) = f_m(y)\left(-\left(\frac{y-\xi}{\sigma}\right)\frac{1}{\sigma} - \sum_{j=1}^{m}\left[a_j\sqrt{2}\sin\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right) j\pi\phi\left(\frac{y-\xi}{\sigma}\right)\frac{1}{\sigma}\right]\right) \tag{2.9}$$

$$\sim \frac{y-\xi}{\sigma} + \sqrt{2}\pi\phi\left(\frac{y-\xi}{\sigma}\right)\sum_{j=1}^{m}\left[a_j j\sin\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right] = 0, \tag{2.10}$$

where $\sim$ means that the two has equivalent roots. It is hard to find the solution to this equation analytically, and it probably has several, so a numerical approach is preferred. For the partial derivatives, it is obtained using implicit differentiation for the points where $f'_m(y) = 0$. For ξ we have that

$$0 = \frac{\frac{dy}{d\xi}-1}{\sigma} + \sqrt{2}\pi\phi'\left(\frac{y-\xi}{\sigma}\right)\frac{\frac{dy}{d\xi}-1}{\sigma}\sum_{j=1}^{m}\left[a_j j\sin\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right]$$

$$+\sqrt{2}\pi\phi\left(\frac{y-\xi}{\sigma}\right)\sum_{j=1}^{m}\left[a_j j\cos\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right) j\pi\phi\left(\frac{y-\xi}{\sigma}\right)\frac{\frac{dy}{d\xi}-1}{\sigma}\right],$$

which gives that $dy/d\xi = 1$. This is natural since ξ is a pure location parameter. For σ we have that

$$0 = \frac{\frac{dy}{d\sigma}\sigma - (y-\xi)}{\sigma^2} + \sqrt{2}\pi\phi'\left(\frac{y-\xi}{\sigma}\right)\frac{\frac{dy}{d\sigma}\sigma-(y-\xi)}{\sigma^2}\sum_{j=1}^{m}\left[a_j j\sin\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right]$$

$$+\sqrt{2}\pi\phi\left(\frac{y-\xi}{\sigma}\right)\sum_{j=1}^{m}\left[a_j j\cos\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right) j\pi\phi\left(\frac{y-\xi}{\sigma}\right)\frac{\frac{dy}{d\sigma}\sigma-(y-\xi)}{\sigma^2}\right],$$

which gives that $dy/d\sigma = (y-\xi)/\sigma$. For a_k, let

$$\mathcal{P} = \sum_{j=1}^{m} \left[a_j j \sin\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right], \tag{2.11}$$

then

$$0 = \frac{\frac{\partial y}{\partial a_k}}{\sigma} + \sqrt{2}\pi\phi'\left(\frac{y-\xi}{\sigma}\right)\frac{\frac{\partial y}{\partial a_k}}{\sigma}\mathcal{P} + \sqrt{2}\pi\phi\left(\frac{y-\xi}{\sigma}\right)\frac{\partial \mathcal{P}}{\partial a_k}, \tag{2.12}$$

where

$$\begin{aligned}
\frac{\partial \mathcal{P}}{\partial a_k} &= \frac{\partial}{\partial a_k}\sum_{j=1, j\neq k}^{m}\left[a_j j \sin\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right] + \frac{\partial}{\partial a_k}\left[a_k k \sin\left(k\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right] \\
&= \sum_{j=1, j\neq k}^{m}\left[a_j j \cos\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right) j\pi\phi\left(\frac{y-\xi}{\sigma}\right)\right]\frac{\frac{\partial y}{\partial a_k}}{\sigma} \\
&+ k\sin\left(k\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right) + \left[a_k k \cos\left(k\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right) k\pi\phi\left(\frac{y-\xi}{\sigma}\right)\frac{\frac{\partial y}{\partial a_k}}{\sigma}\right] \\
&= \sum_{j=1}^{m}\left[a_j j \cos\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right) j\pi\phi\left(\frac{y-\xi}{\sigma}\right)\right]\frac{\frac{\partial y}{\partial a_k}}{\sigma} + k\sin\left(k\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right).
\end{aligned}$$

All this gives that

$$\frac{\partial y}{\partial a_k} = \frac{-k\sqrt{2}\pi\phi\left(\frac{y-\xi}{\sigma}\right)\sin\left(k\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)}{\left(\frac{1}{\sigma} + \sqrt{2}\pi\phi'\left(\frac{y-\xi}{\sigma}\right)\frac{\mathcal{P}}{\sigma} + \sqrt{2}\pi\phi\left(\frac{y-\xi}{\sigma}\right)\sum_{j=1}^{m}\left[a_j j\cos\left(j\pi\Phi\left(\frac{y-\xi}{\sigma}\right)\right)j\pi\phi\left(\frac{y-\xi}{\sigma}\right)\right]\frac{1}{\sigma}\right)},$$

which after some tiding up, and substituting $\epsilon = (y-\xi)/\sigma$ gives

$$\frac{\partial y}{\partial a_k} = \frac{-k\sigma\pi\phi(\epsilon)\sin(k\pi\Phi(\epsilon))}{\frac{1}{\sqrt{2}} + \pi\phi'(\epsilon)\mathcal{P} + (\pi\phi(\epsilon))^2\sum_{j=1}^{m}\left[a_j j^2\cos(j\pi\Phi(\epsilon))\right]}. \tag{2.13}$$

This leads to the following result

Proposition 2.3.1 *For the mode, y_0, of* (2.3) *we have that*

$$\frac{\partial \mu}{\partial \boldsymbol{\theta}} = \begin{pmatrix} 1 \\ \epsilon \end{pmatrix} \tag{2.14}$$

while the general form for $\frac{\partial \mu}{\partial a_k}$ is given in (2.13)*. The narrow model estimates, when $\boldsymbol{a} = \mathbf{0}$, it is simplified to*

$$\frac{\partial \mu}{\partial \boldsymbol{\theta}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.15}$$

since $y = \xi$. For $\boldsymbol{a}$ it is

$$\frac{\partial \mu}{\partial a_k} = -k\sigma\sqrt{\pi}\sin\left(\frac{k\pi}{2}\right). \tag{2.16}$$

Notice that $\sin(u) = 0$ for integer multiples of π, which happens when k is even, so the partial derivative is zero for even k.

2.3.2 Omega for Log Density Estimation

The other focus parameter in discussion is the log density. Let

$$\hat{\mu} = \log\left[f_m(y, \hat{\xi}, \hat{\sigma}, \hat{\boldsymbol{a}})\right]$$

for an arbitrary point y. The log density of (2.3) is

$$\log(f(y)) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\left(\frac{y-\xi}{\sigma}\right)^2 - \log\sigma + \sum_{k=1}^{m} a_k \psi_k\left(\Phi\left(\frac{y-\xi}{\sigma}\right)\right) - \log K_m(\boldsymbol{a})$$

$$= -\frac{1}{2}\log 2\pi - \frac{1}{2}\varepsilon^2 - \log\sigma + \sum_{k=1}^{m}\sqrt{2}a_k\cos\left(k\pi\Phi\left(\varepsilon\right)\right) - \log k_m(\boldsymbol{a})$$

where $\varepsilon = \left(\frac{y-\xi}{\sigma}\right)$. We have that

$$\frac{d}{d\xi}[\varepsilon] = -\frac{1}{\sigma}, \qquad \frac{d}{d\sigma}[\varepsilon] = -\frac{\varepsilon}{\sigma},$$

which leads to

$$\frac{\partial \mu}{\partial \xi} = \frac{\varepsilon}{\sigma} + \sqrt{2}\sum_{k=1}^{m} a_k \sin\left(k\pi\Phi(\varepsilon)\right) k\pi\phi(\varepsilon)\frac{1}{\sigma}$$

$$\frac{\partial \mu}{\partial \sigma} = \frac{\varepsilon - 1}{\sigma} + \sqrt{2}\sum_{k=1}^{m} a_k \sin\left(k\pi\Phi(\varepsilon)\right) k\pi\phi(\varepsilon)\frac{\varepsilon}{\sigma}$$

For the $\boldsymbol{a}$ vector, an integral is needed. We get that

$$\frac{\partial \mu}{\partial a_k} = \sqrt{2}\cos\left(k\pi\Phi\left(\varepsilon\right)\right) - \frac{d}{da_k}\log k_m(\boldsymbol{a})$$

$$= \sqrt{2}\cos\left(k\pi\Phi\left(\varepsilon\right)\right) - \frac{d}{da_k}\int_0^1 \exp\left(\sum_{j=1}^{m} a_j\sqrt{2}\cos(j\pi\phi(u)\right) du\frac{1}{k_m(\boldsymbol{a})}$$

$$= \sqrt{2}\cos\left(k\pi\Phi\left(\varepsilon\right)\right) - \int_0^1 \exp\left(\sum_{j=1}^{m} a_j\sqrt{2}\cos(j\pi\phi(u)\right)\sqrt{2}\cos(j\pi\phi(u)du\frac{1}{k_m(\boldsymbol{a})}.$$

2.4 Ficology

This model is quite 'nice' when it comes to ficology at the narrow model. We have that

$$J_{00}^{-1} = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \frac{\sigma^2}{2} \end{pmatrix}$$

$$\tau_0^2 = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \sigma^2 & 0 \\ 0 & \frac{\sigma^2}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \sigma^2$$

$$\boldsymbol{\omega} = \begin{pmatrix} c_1 & d_1 \\ c_2 & d_2 \\ \vdots & \vdots \\ c_q & d_q \end{pmatrix} \begin{pmatrix} \sigma^2 & 0 \\ 0 & \frac{\sigma^2}{2} \end{pmatrix} \begin{pmatrix} \frac{\partial \mu}{\partial \xi} \\ \frac{\partial \mu}{\partial \sigma} \end{pmatrix} - \frac{\partial \mu}{\partial \boldsymbol{a}} = \sigma^2 \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_q \end{pmatrix} - \frac{\partial \mu}{\partial \boldsymbol{a}}.$$

Since $c_i = 0$ for even i, we get for an arbitrary focus parameter μ that

$$\boldsymbol{\omega} = \sigma^2 \begin{pmatrix} c_1 - \frac{\partial \mu}{\partial a_1} \\ -\frac{\partial \mu}{\partial a_2} \\ c_3 - \frac{\partial \mu}{\partial a_3} \\ \vdots \\ c_q - \frac{\partial \mu}{\partial a_q} \end{pmatrix}.$$

2.4.1 Tolerance Bands and Ellipses

When comparing two models, a narrow and a wide, the narrow model is better whenever

$$\boldsymbol{\omega}^t \boldsymbol{\delta} \boldsymbol{\delta}^t \boldsymbol{\omega} + \tau_0^2 < \tau_0^2 + \boldsymbol{\omega}^t Q \boldsymbol{\omega}.$$

Withdrawing τ_0^2 on both sides and rooting gives

$$|\boldsymbol{\omega}^t \boldsymbol{\delta}| < (\boldsymbol{\omega}^t Q \boldsymbol{\omega})^{\frac{1}{2}},$$

which solved for $\boldsymbol{\delta}$ is an infinite band in $\mathbb{R}^q$. Claeskens and Hjort (2008, Thm. 5.3) tells that the narrow model is better than the wide for all foci when

$$\boldsymbol{\delta}^t Q^{-1} \boldsymbol{\delta} \leq 1.$$

The solution for $\boldsymbol{\delta}$ to this equations forms a hyperellipse in $\mathbb{R}^q$. Assume that

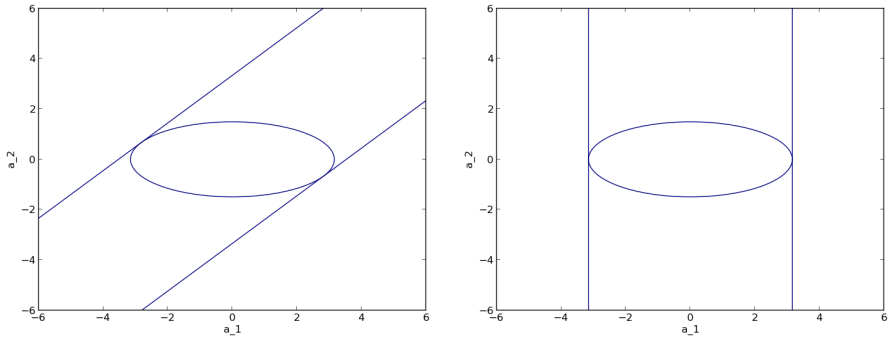


Figure 2.7: The band where the narrow model beats the model with $m = 2$ for the log density at $y = \xi$ to the left, and for the mode to the right, along with the ellipsoid where the narrow model is better for every focus parameter.

$q = 2$. For the mode and log density we get that

$$\boldsymbol{\omega}_{mode} = \begin{pmatrix} c_1 - \frac{\partial \mu}{\partial a_1} \\ 0 \end{pmatrix} = \begin{pmatrix} -0.9475 - (-1.7725) \\ 0 \end{pmatrix} = \begin{pmatrix} 0.8250 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\omega}_{logdens} = \begin{pmatrix} c_1 - \frac{\partial \mu}{\partial a_1} \\ \frac{\partial \mu}{\partial a_2} \end{pmatrix} = \begin{pmatrix} -0.9475 \\ 1 \end{pmatrix}$$

We also have that

$$\boldsymbol{c} = (-0.9475, 0.0000), \qquad \boldsymbol{d} = (0.0000, 1.0499),$$

which gives that

$$Q = \begin{pmatrix} 9.9521 & 0.0000 \\ 0.0000 & 2.2261 \end{pmatrix}.$$

The tolerance bands for the mode, and for log density estimation at $y = \xi$ are shown along with the ellipsoid in figure 2.7. An interesing feature of the rightward plot in the figure, is that a_2 can be anything, as long as a_1 is within certain limits, when estimating the mode.

2.5 Drawing Random Variables

One of many random number generating algorithms is the acceptance rejection algorithm. This version is presented in Rubenstein and Kroese (2008, p. 56), and is as follows:

1. Draw $Y_{prop} \sim h(y)$
2. Draw $u \sim U[0, c \cdot h(Y_{prop})]$
3. If $u \leq f(Y_{prop})$, keep Y_{prop} as a random variable from $f(y)$

The distribution $h(y)$ is called the proposal distribution. The constant c must satisfy $c \cdot h(y) \geq f_m(y)$ for all y. It can be shown that the probability for a proposed random variable to be accepted is $1/c$.

For the density f_m, the proposal distribution is chosen to be the normal distribution $\mathcal{N}(\xi, \sigma)$. The first step is to find a $c \geq \max_y f_m(y)/h(y)$:

$$\max_x \left\{ \phi\left(\frac{y-\xi}{\sigma}\right) \frac{1}{\sigma} \exp\left(\sum_{j=1}^{m} a_j \psi_j\left(\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right) \frac{1}{k_m(\boldsymbol{a})} \left[\phi\left(\frac{y-\xi}{\sigma}\right)\frac{1}{\sigma}\right]^{-1} \right\}$$

$$= \max_x \left\{ \exp\left(\sum_{j=1}^{m} a_j \psi_j\left(\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right) \frac{1}{k_m(\boldsymbol{a})} \right\}$$

$$\leq \exp\left(\sqrt{2}\sum_{j=1}^{m} |a_j|\right) \frac{1}{k_m(\boldsymbol{a})},$$

so one possible choice of c is

$$c = \exp\left(\sqrt{2}\sum_{j=1}^{m} |a_j|\right) \frac{1}{k_m(\boldsymbol{a})}.$$

For our specific case we get the algorithm

1. Draw $Y_{prop} \sim \mathcal{N}(\xi, \sigma)$
2. Generate $u \sim \mathcal{U}[0, c]$
3. Keep Y_{prop} if
$$u < \exp\left(\sum_{j=1}^{m} a_j \psi_j\left(\Phi\left(\frac{y-\xi}{\sigma}\right)\right)\right).$$

2.6 Example with Stepwise Instructions

The data in this example are from a study of factors leading to low birthweights, and is introduced in Hosmer and Lemeshow (1999). The dataset contains data for $n = 189$ babies, and includes 11 covariates[2]. In this analysis we

[2]See http://www.econ.kuleuven.ac.be/public/ndbaf45/modelselection/

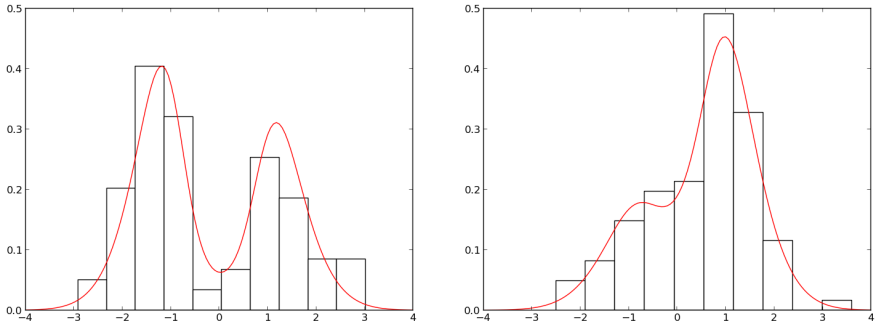


Figure 2.8: Histograms with two sets of $n = 100$ random variables each, drawn from (2.3) plotted with corresponding density function. The parameters used are $\xi = 0$ and $\sigma = 1$ for both figures, and $(a_1, a_2) = (0.1, 0.99)$ to the left and $(a_1, a_2) = (-0.4, 0.4)$ to the right.

will only look at the babyweights, and try to estimate the mode and the underlying density. Fitting the model with $m = 4$ parameters in the log expansion, gives parameter estimates

$$\begin{pmatrix} \hat{\xi} \\ \hat{\sigma} \end{pmatrix} = \begin{pmatrix} 2.5677 \\ 0.7528 \end{pmatrix}, \qquad \begin{pmatrix} \hat{a_1} \\ \vdots \\ \hat{a_4} \end{pmatrix} = \begin{pmatrix} -0.4931 \\ 0.0043 \\ -0.2095 \\ -0.0248 \end{pmatrix},$$

which gives that

$$D_n = \sqrt{n}(\hat{\boldsymbol{a}} - \mathbf{0}) = \begin{pmatrix} -6.7796 \\ 0.0589 \\ -2.8807 \\ -0.3409 \end{pmatrix}.$$

For the narrow model, the parameter estimates $(\hat{\xi}_0, \hat{\sigma}_0) = (2.9443, 0.7271)$.

2.6.1 Narrow vs. Wide Estimated FIC

In this thesis, two main versions of the FIC are used, namely wide and narrow estimated FIC. Under the sequence of models $f_n = f_m(y; \boldsymbol{\theta}, \gamma_0 + \delta/\sqrt{n})$, the information matrix J, and the $\boldsymbol{\omega}$ could be estimated using either the narrow model parameter estimates $(\hat{\boldsymbol{\theta}}_0, \gamma_0)$, or the wide model estimates $(\hat{\boldsymbol{\theta}}, \hat{\gamma})$. The former gives some robustness.

Another thing is that the narrow model could be any sub model of the wide. In this thesis we have presented the narrow model as the normal distribution with parameter (ξ, σ). Generally one could also use one or more a's in the narrow

model of the analysis. For example, the narrow model could have parameters (ξ, σ, a_1, a_2) while the wide model har parameters $(\xi, \sigma, a_1, a_2, a_3, a_4)$. In that case numerical methods would be used for both narrow and wide estimated FIC.

2.6.2 Modehunting with Narrow Estimated FIC

In the mode hunt, the focus parameter μ is

$$\mu(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \arg\max_y f_m(y).$$

Calculations show that at $\boldsymbol{a} = \mathbf{0}$,

$$\frac{\partial \mu}{\partial(\hat{\xi}_0, \hat{\sigma}_0)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \frac{\partial \mu}{\partial \boldsymbol{a}} = \left\{ -k\sigma \sin\left(\frac{k\pi}{2}\right) \right\}_{k=1}^{4} = \begin{pmatrix} -1.2887 \\ 0.0000 \\ 3.8662 \\ 0.0000 \end{pmatrix}.$$

Combined with the narrow estimate of J from (2.7), with $\hat{\sigma}_0$ plugged in for σ, we get that

$$\hat{Q} = \begin{pmatrix} 22.1434 & 0.0000 & 5.3660 & 0.0000 \\ 0.0000 & 2.9052 & 0.0000 & 1.0272 \\ 5.3660 & 0.0000 & 2.3618 & 0.0000 \\ 0.0000 & 1.0272 & 0.0000 & 1.5539 \end{pmatrix},$$

and that

$$\hat{\boldsymbol{\omega}} = \begin{pmatrix} 0.5991 \\ 0.0000 \\ -4.0412 \\ 0.0000 \end{pmatrix}.$$

Calculating FIC Scores

When calculating the FIC scores we do four calculations for each model. Assume as an example that we want to estimate FIC for $m = 2$, that is with parameters (ξ, σ, a_1, a_2). The projection matrix π_2 is then

$$\pi_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

This gives that

$$\hat{Q}_2 = (\pi_2 \hat{Q} \pi_2^t)^{-1}$$

$$\hat{G}_2 = \pi_2^t \hat{Q}_2 \pi_2 \hat{Q}^{-1},$$

where $\hat{Q}_2$, for reference, is a $2 \times 2 = |S| \times |S|$-matrix, and $\hat{Q}_2$ is $4 \times 4 = q \times q$. The FIC variance and bias are

$$\hat{\text{var}}(\hat{\mu}_2) = \hat{\tau}_0^2 + (\pi_2 \hat{\omega})^t \hat{Q}_2 (\pi_2 \hat{\omega})$$

$$\hat{\text{bias}}^2(\hat{\mu}_2) = \hat{\omega}^t (I_q - \hat{G}_2)(D_n D_n^t - \hat{Q})(I_q - \hat{G}_2)^t \hat{\omega}.$$

This is repeated for every sub model in the analysis.

The FIC Table

Calculating the FIC scores for models $S = 0, \ldots, 4$, and $\pi_0 = 0$, gives the table

Model	st.dev	bias	rFIC	$\hat{\mu}$
m = 0	0.7271	6.0754	6.1188	2.9443
m = 1	2.0251	6.5297	6.8365	3.0176
m = 2	2.0251	6.5297	6.8365	3.2505
m = 3	4.5896	0.0000	4.5896*	3.2769
m = 4	4.5896	0.0000	4.5896	3.2697

In the table, st.dev $= \sqrt{\widehat{\text{var}}(\hat{\mu}_S)}$ and

$$\text{bias} = \text{sgn}\left[\widehat{\text{bias}}^2(\hat{\mu}_S)\right] \cdot \sqrt{|\widehat{\text{bias}}^2(\hat{\mu}_S)|},$$

while rFIC, root FIC, is

$$rFIC_S = \sqrt{\widehat{\text{var}}(\hat{\mu}_S) + \max\left\{\widehat{\text{bias}}^2(\hat{\mu}_S), 0\right\}}.$$

One interesting feature is that both the bias and the variance of the even numbered models (except 0) are exactly the same as the models with one less index. The scores of models 1 and 2 are the same, and so are 3 and 4. This is because all the even numbered components in ω is zero. This is consistent the tolerance band for the mode presented in figure 2.7, where the even numbered parameters are basically without restriction in the question of narrow versus wide.

2.6.3 Modehunting with Wide Estimated FIC

The difference between narrow estimated FIC and wide estimated FIC, is how estimates for ω and J are obtained. First we use numerical methods to obtain the empirical $\hat{J}$ matrix

$$\hat{J} = -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial^2 \ell(y_i; \zeta)}{\partial \zeta \partial \zeta^t}.$$

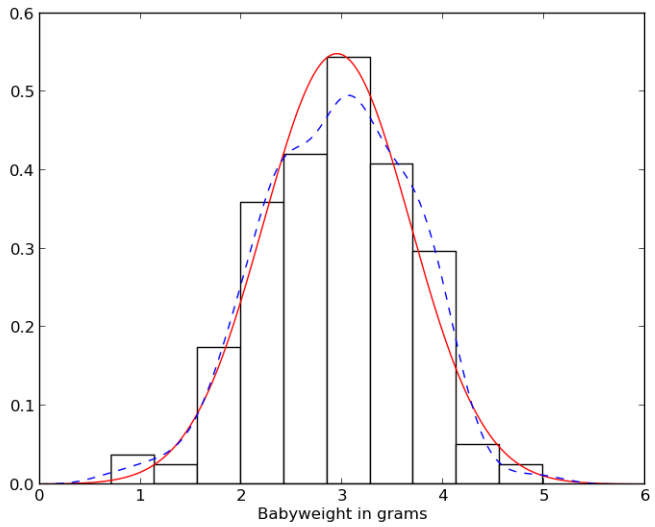


Figure 2.9: Histogram of the Smallbabies data with normal fit(—), and kernel density estimate (- -).

where $\boldsymbol{\zeta} = (\hat{\xi}, \hat{\sigma}, \hat{\boldsymbol{a}})$. Computations show that the $\hat{Q}$-matrix is

$$\hat{Q} = \begin{pmatrix} 36.9043 & 13.3303 & 9.6968 & -9.6042 \\ 13.3303 & 6.8170 & 3.5831 & -2.7400 \\ 9.6968 & 3.5831 & 3.9197 & -2.4103 \\ -9.6042 & -2.7399 & -2.4103 & 4.3893 \end{pmatrix}. \tag{2.17}$$

This estimate of Q is quite different from the narrow estimate, even though $\hat{\boldsymbol{a}}$ is quite close to $\mathbf{0}$. The differences comes from the estimate of J, which changes much from the narrow model to the wide.
Plugging the wide model fit into the formulas in proposition 2.3.1, we get

$$\frac{\partial \mu}{\partial(\hat{\xi}, \hat{\sigma})} = \begin{pmatrix} 1.0000 \\ 0.9325 \end{pmatrix}, \quad \frac{\partial \hat{\boldsymbol{a}}}{\partial \mu} = \begin{pmatrix} -0.1934 \\ 0.6589 \\ -1.1035 \\ 1.1886 \end{pmatrix}, \quad \hat{\boldsymbol{\omega}} = \begin{pmatrix} -0.2725 \\ -0.3100 \\ 0.7855 \\ -0.7606 \end{pmatrix}.$$

Using the same procedure as earlier we get a FIC table

Model	s	bias	rFIC	$\hat{\mu}$
m=0	0.6971	-1.5118	0.6971*	2.9443
m=1	0.8921	0.4699	1.0083	3.0176
m=2	1.4415	-0.8465	1.4415	3.2505
m=3	1.4463	-0.8315	1.4463	3.2769
m=4	1.6739	0.0000	1.6739	3.2697

which favours the narrow model.

2.6.4 Density Estimation with Narrow Estimated Average-FIC

When it comes to density estimaton, we actually select model for the log density, because of mathematical convenience. The results will be the same. The focus parameter is in this case

$$\mu = \log f_m(y; \xi, \sigma, \boldsymbol{a}),$$

at a given point y. All of the input variables in the Average-FIC formulas are calculated previously, except A. One method is to use numerical integration. Recall from chapter one that the B matrix is defined as

$$B = \int_{-\infty}^{\infty} \begin{pmatrix} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mu}{\partial \gamma} \end{pmatrix} \begin{pmatrix} \frac{\partial \mu}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mu}{\partial \gamma} \end{pmatrix}^t dW(u) = \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix},$$

which is $(p+q)^2$ weighted integrals. For this example we use numerical integration, and integrate over $(a, b) = (0.7, 5.0)$, which covers the sample range. The estimate $\hat{B}$ of B is

$$\hat{B} = \begin{pmatrix} 5.5562 & -1.9080 & -2.7465 & -0.2522 & -2.1316 & -0.2522 \\ -1.9080 & 20.3780 & 0.6618 & 4.3846 & 0.6618 & 4.0157 \\ -2.7465 & 0.6618 & 1.5077 & 0.0876 & 0.9117 & 0.0876 \\ -0.2522 & 4.3846 & 0.0876 & 1.4040 & 0.0876 & 0.8543 \\ -2.1316 & 0.6618 & 0.9117 & 0.0876 & 1.3466 & 0.0876 \\ -0.2522 & 4.0157 & 0.0876 & 0.8543 & 0.0876 & 1.3075 \end{pmatrix},$$

which combined with the narrow estimate $\hat{J}$ of J from (2.7) gives

$$\hat{A} = \begin{pmatrix} 0.3619 & 0.1632 & -0.3683 & 0.0482 \\ 0.1632 & 1.0248 & -0.0816 & 0.0197 \\ -0.3683 & -0.0816 & 0.7707 & -0.0240 \\ 0.0482 & 0.0197 & -0.0240 & 0.5177 \end{pmatrix}.$$

The formulas for the Average-FIC are

$$\begin{aligned} \hat{I}(S) &= Tr\left((I_q - \hat{G}_S)(D_n D_n^t - Q)(I_q - \hat{G}_S)^t \hat{A}\right) \\ \hat{I}I(S) &= Tr\left(\pi_S^t Q_S \pi_S \hat{A}\right) \\ AFIC(S) &= \max\left\{\hat{I}(S), 0\right\} + \hat{I}I(S), \end{aligned}$$

where π_S is a projection matrix as defined earlier. This is repeated for every submodel in the analysis. The resulting table is similar to that of FIC.

Model	st.dev	bias	rAFIC
m=0	0.0000	-0.9613	0.0000
m=1	1.8978	2.2568	2.9487
m=2	2.4255	2.5753	3.5376
m=3	2.8570	0.0000	2.8570
m=4	3.1150	0.0000	3.1150

Here

$$\text{st.dev} = \sqrt{\hat{I}I(S)}$$

$$\text{bias} = \text{sgn}(\hat{I}(S)) \cdot \sqrt{|\hat{I}(S)|}$$

$$rAFIC_S = \sqrt{\max\left\{\hat{I}(S), 0\right\} + \hat{I}I(S)}.$$

2.6.5 Density Estimation with Wide Estimated Average-FIC

Plugging in the wide parameter estimates for $\frac{\partial \mu}{\partial(\hat{\xi},\hat{\sigma})}$ and $\frac{\partial \mu}{\partial \hat{\boldsymbol{a}}}$, gives the following estimate for B

$$\hat{B} = \begin{pmatrix} 5.6072 & 4.1614 & -2.5846 & 0.4997 & -2.0129 & 0.9836 \\ 4.1614 & 19.4264 & -0.0836 & 3.7549 & -1.2208 & 3.6078 \\ -2.5846 & -0.0836 & 1.5369 & 0.0216 & 0.8406 & -0.0206 \\ 0.4997 & 3.7549 & 0.0216 & 1.1966 & -0.1219 & 0.7040 \\ -2.0129 & -1.2208 & 0.8406 & -0.1219 & 1.2867 & -0.1512 \\ 0.9836 & 3.6078 & -0.0206 & 0.7040 & -0.1512 & 1.2334 \end{pmatrix},$$

which combined with the wide estimate of J gives

$$\hat{A} = \begin{pmatrix} 0.1462 & -0.1486 & -0.1582 & 0.0607 \\ -0.1486 & 0.7656 & -0.0902 & 0.2201 \\ -0.1582 & -0.0902 & 0.9831 & 0.4041 \\ 0.0607 & 0.2201 & 0.4041 & 0.9201 \end{pmatrix}.$$

Just as with $\hat{Q}$, the wide and narrow estimates of A are different, even though $\hat{\boldsymbol{a}}$ is close to zero. The estimates of B are more similar but there are differences there as well. This tells that the narrow and wide estimated Average-FIC could be different.

Model	s	bias	rAFIC
m=0	0.0000	1.8739	1.8739
m=1	0.7811	1.7634	1.9287
m=2	1.6521	2.9437	3.3757
m=3	2.1979	0.0000	2.1979
m=4	2.5517	0.0000	2.5517

which again favours the narrow model.

2.6.6 Testing for Significant Parameters

Assume the hypotheses $H_0 : a_k = 0$ versus $H_a : a_k \neq 0$ for each a. Asymptotically $\text{cov}[\sqrt{n} \cdot \hat{\boldsymbol{a}}] = Q$, which means that under the null hypotheses, we have for parameter a_k that

$$Z_n = \sqrt{n}\frac{\hat{a}_k}{\sqrt{\hat{Q}_{k,k}}} \rightarrow_D \mathcal{N}(0,1).$$

Assume the wide fit with $m = 4$, and the wide estimated $\hat{Q}$ matrix in (2.17). A Wald test resulted in the following table:

Param	Est.	St.Dev.	Z-val	$P(>\|Z\|)$
a_1	-0.4931	2.6844	-0.1837	0.8542
a_2	0.0043	0.4959	0.0086	0.9931
a_3	-0.2095	0.2851	-0.7349	0.4624
a_4	-0.0248	0.3193	-0.0777	0.9381

Another test one could use is a χ^2 test to test the four parameters in $\hat{\boldsymbol{a}}$ simultaneously.The test estimator is

$$Z^2 = n\hat{\boldsymbol{a}}^t\hat{Q}^{-1}\hat{\boldsymbol{a}},$$

which is approximately χ^2_4-distributed under the null hypotheses $H_0 : \boldsymbol{a} = \boldsymbol{0}$. Calculations show that $Z^2 = 10.7968$, which at 4 degrees of freedom gives $P(\chi^2_4 > Z^2) = 0.0289$. This indicates that the vector $\boldsymbol{a}$ is significantly different from the zero vector. While the Q-Q plot in figure 2.10 and the Wald test strengthens the theory that the dataset is normal, the χ^2 test doubts it.

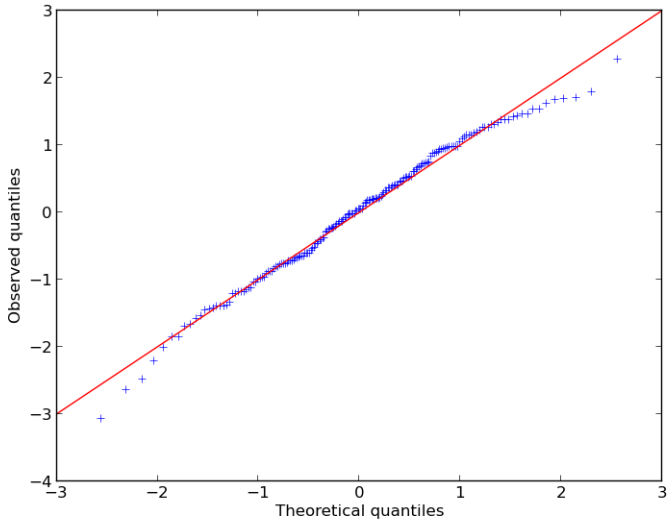


Figure 2.10: Q-Q plot of the babyweight dataset to check for normality.

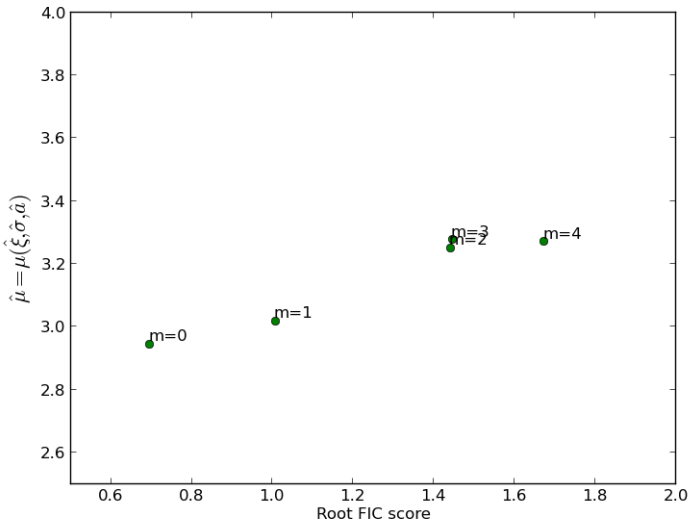


Figure 2.11: FIC plot for the wide FIC analysis. The root FIC scores are on the x-axis, while the estimated focus parameter for the corresponding model is on the y-axis. Points further left has lower FIC score.

Chapter 3

Mode Hunting with FIC

3.1 Properties of a Test Class

The scheme developed in the previous section needs verification. For the test, introduce the normal mixture class of distributions

$$g(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p}) = \sum_{i=1}^{k} p_i \phi\left(\frac{y-\mu_i}{\tau_i}\right)\frac{1}{\tau_i}, \tag{3.1}$$

where $\sum_{i=1}^{n} p_i = 1$. Since the integral of a sum is the sum of the integrals, the cumulative distribution is

$$G(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p}) = \sum_{i=1}^{k} p_i \Phi\left(\frac{y-\mu_i}{\tau_i}\right).$$

Furthermore, computing the moments can be done through moment generating functions. Since we are dealing with sums of integrals, and the moment generating function of the normal distribution is (Lehmann (1999, p. 582))

$$M_\phi(t) = e^{\mu t+\tau^2 t^2},$$

we get that the moment generating function of g is

$$M_g(t) = \sum_{i=1}^{k} p_i \exp\left(\mu_i t + \frac{1}{2}\tau_i^2 t^2\right).$$

Differentiating with respect to t gives

$$M_g'(t) = \sum_{i=1}^{k} p_i \exp\left(\mu_i t + \frac{1}{2}\tau_i^2 t^2\right)\left(\mu_i + \tau_i^2 t\right)$$

$$M_g''(t) = \sum_{i=1}^{k} p_i \exp\left(\mu_i t + \frac{1}{2}\tau_i^2 t^2\right)\left(\left(\mu_i + \tau_i^2 t\right)^2 + \tau_i^2\right),$$

which gives that

$$E_g[Y] = \sum_{i=1}^{k} p_i \mu_i$$

$$E_g[Y^2] = \sum_{i=1}^{k} p_i \left(\mu_i^2 + \tau_i^2 \right),$$

and that the variance is

$$\text{var}_g[Y] = E_g[Y^2] - E_g[Y]^2 = \sum_{i=1}^{k} \left[p_i \left(\mu_i^2 + \tau_i^2 \right) - \left(\sum_{i=1}^{k} p_i \mu_i \right)^2 \right].$$

The kurtosis and skewness can be found in the same way.

3.1.1 Drawing Random Variables

Drawing random variables Y from this distribution can be done efficiently with the composition method, described in Rubenstein and Kroese (2008, p. 51). Mixture distributions has general form

$$g(y) = \sum_{i=1}^{k} p_i g_i(y), \qquad \sum_{i=1}^{k} p_i = 1,$$

and the algorithm goes as follows

1. Draw one discrete random variable X, such that $P(X = i) = p_i$, for $i = 1, \ldots, k$

2. Given $X = i$, draw the random variable Y from g_i

In this case, we have that $g_i = \mathcal{N}(\mu_i, \tau_i)$. So the algorithm consists of drawing one discrete random variable, and one normal random variable. Two examples of random samples are presented in figure 3.1.

3.1.2 Test Parameters

For the tests, we use six distributions from the normal mixture, with different parameters. The parameters, along with mode, mean and variance, are presented in table 3.1, and are illustrated in figure A.1. The test distributions range from unimodal to trimodal.

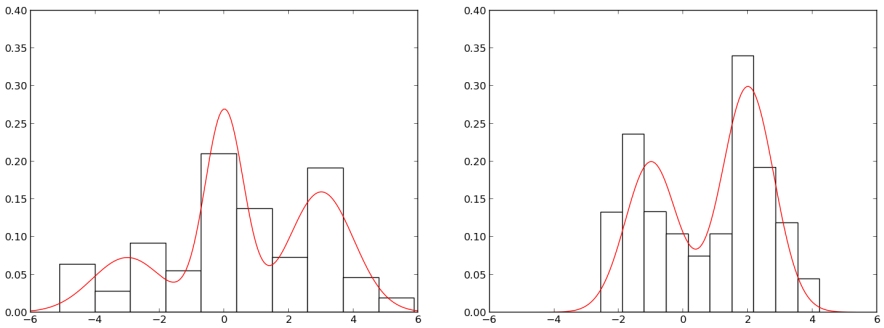


Figure 3.1: Two histograms of samples of $n = 100$ random variables, from distributions (3.1) with corresponding density function. Variables are generated with the algorithm i section 3.1.1, with parameters from test distribution 3 (left) and 5 (right), as described in table 3.1.

#	μ	τ	p	Mode	Mean	Variance
1	0	1	1	0.0	0.0	1.0000
2	$(-1, 2)$	$(1.1, 0.6)$	$(0.4, 0.6)$	2.0	0.8	2.9400
3	$(-3, 0, 3)$	$(1.1, 0.6, 1)$	$(0.2, 0.4, 0.4)$	0.0	0.6	5.8260
4	$(-3, 0, 3)$	$(0.8, 0.6, 0.8)$	$(0.2, 0.6, 0.2)$	0.0	0.0	4.0720
5	$(-1, 2)$	$(0.8, 0.8)$	$(0.4, 0.6)$	2.0	0.8	2.8000
6	$(-1, 2)$	$(0.8, 0.8)$	$(0.45, 0.55)$	2.0	0.65	2.8675

Table 3.1: Parameters of the six test distributions from (3.1) that are used as reference in the tests. The test distributions are illustrated in table A.1.

3.1.3 Estimating Least False Parameters

We know from earlier that the maximum likelihood estimate is the empirical minimizer of the Kullback-Leibler divergence, so obtaining the least false parameters of $f_m(y;\xi,\sigma,\boldsymbol{a})$ when estimating $g(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p})$ is done by minimizing

$$\hat{D}_{KL} = \int_I g(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p}) \log \frac{g(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p})}{f_m(y;\xi,\sigma,\boldsymbol{a})} dy \tag{3.2}$$

numerically for some appropriate interval I, with respect to $(\xi,\sigma,\boldsymbol{a})$.

The interval $I = (-10,10)$ should suffice in this experiment, since the test distributions are close to zero outside that interval. A table with the least false Kullback-Leibler divergences are in gathered in table A.1. The least false modes are in table A.2, while the computed least false parameters are presented in tables A.2 and A.3.

3.2 Least False Computations

This section is devoted to the test results that can be derived analytically without having to use Monte Carlo simulations.

3.2.1 Approximations

The approximate mean squared error of the kernel estimated mode with Gaussian kernel is

$$\text{mse}\,[\hat{y}_{0,K}] = \left[\frac{h^2 \cdot g^{(3)}(y_0)}{2g^{(2)}(y_0)}\right]^2 + \frac{g(y_0)(\sqrt{2\pi})^{-1}}{nh^3[g^{(2)}(y_0)]^2}, \tag{3.3}$$

with h as in (1.15). For the parametric estimates, we know with theory from chapter 1 that

$$\sqrt{n}(\hat{\boldsymbol{\zeta}} - \boldsymbol{\zeta}_0) \to_D \mathcal{N}\left(0, J^{-1}(\boldsymbol{\zeta}_0)K(\boldsymbol{\zeta}_0)J^{-1}(\boldsymbol{\zeta}_0)\right),$$

where $\boldsymbol{\zeta} = (\xi,\sigma,\boldsymbol{a})$, and $\boldsymbol{\zeta}_0$ is the least false parameter vector. Since we know the partial derivatives of the mode, the least false parameters, and how to estimate J and K, we get that the approximate variance of the estimated mode is

$$\text{var}(\hat{y}_{0,P}) \approx \frac{1}{n}\frac{\partial y_0}{\partial \boldsymbol{\zeta}_0}^t J^{-1}(\boldsymbol{\zeta}_0)K(\boldsymbol{\zeta}_0)J^{-1}(\boldsymbol{\zeta}_0)\frac{\partial y_0}{\partial \boldsymbol{\zeta}_0},$$

where the least false mode is plugged in for y_0. The squared bias is obtained by comparing table A.2 and table 3.1. This gives that the approximate root mean squared error is

$$\text{rmse}(\hat{y}_{0,P}) \approx \sqrt{\text{var}(\hat{y}_{0,P}) + (\hat{y}_0 - y_0)^2}. \tag{3.4}$$

Model	1	2	3	4	5	6
$n = 50$	0.3812	1.2467	1.2325	1.1936	0.7396	0.7783
$n = 200$	0.2832	1.3906	0.9376	0.8868	0.7103	0.6624
$n = 1000$	0.2006	1.9306	0.7155	0.6282	0.5494	0.5169

Table 3.2: Table over the approximate relative root mean squared error of the parametric estimates and the kernel estimate. The m used in each estimation is the one with lowest rmse from the approximations (3.4).

Some of these approximations are put in table 3.2, which is the approximate relative root mean squared error of the parametric estimates and the kernel estimate. The m used in each estimation is the one with lowest rmse from the approximations (3.4).

3.2.2 Asymptotic Effects and Results

Large Sample Bias Effect

The expression for (3.3) for our case, with kernel bandwidth of order $1/7$, can be expressed on the form

$$\text{mse}\,[\hat{y}_{0,K}] = \mathcal{D}_1 n^{-4/7} + \mathcal{D}_2 n^{-4/7},$$

for constants $\mathcal{D}_1$ and $\mathcal{D}_2$. Assume a parametric mode estimate with bias $\mathcal{C}_\infty$, and variance $\mathcal{C}_2/n$, then the mean squared error is on the form

$$\text{mse}[\hat{y}_{0,P}] = \mathcal{C}_1 + \mathcal{C}_2 n^{-1}.$$

This gives that the approximate relative mean squared error is

$$\frac{\mathcal{C}_1 + \mathcal{C}_2 n^{-1}}{(\mathcal{D}_1 + \mathcal{D}_2) n^{-4/7}} = \frac{\mathcal{C}_1}{\mathcal{D}_1 + \mathcal{D}_2} n^{4/7} + \frac{\mathcal{C}_2}{\mathcal{D}_1 + \mathcal{D}_2} n^{-3/7}.$$

This blows up as $n \to \infty$ if $\mathcal{C}_1 > 0$. If there is no bias, the relative mean squared error will go to zero in order $n^{-3/7}$.

Relative Efficiency

If we look at just the variance, we know that the asymptotic relative efficiency is of order $\mathcal{O}(n^{-3/7})$ in favour of the parametric estimates. However, for low n the coefficients $\mathcal{C}_2$ and $\mathcal{D}_2$ are just as important. We have for test distribution 6 that

$$\frac{g(\theta)}{[g''(\theta)]^2} V = 0.0637, \qquad \text{var}[\hat{y}_0] = 1.3052,$$

so the $ARE(n) = 20.4887n^{-3/7}$, which is less than 1 whenever n is larger than 1149. So the parametric estimate is asymptotically efficient, but that doesn't matter as long as n is small to moderately large. Calculations show that the standard deviation for the kernel estimates, and the test distributions for a sample size of $n = 10$ are

$$\left\{ \frac{g(y_0)}{[g''(y_0)]^2} V \right\}_{M=1}^{6} = \begin{pmatrix} 0.3162 \\ 0.1473 \\ 0.0751 \\ 0.6381 \\ 0.1060 \\ 0.1180 \end{pmatrix}, \qquad \text{var}_M[X] = \begin{pmatrix} 0.3162 \\ 0.5420 \\ 0.7632 \\ 0.6381 \\ 0.5292 \\ 0.5355 \end{pmatrix},$$

which shows that the kernel estimate has a head start for low n.

Results

For this part, we have that the approximate relative mean squared error can be written on the form

$$\frac{\mathcal{C}_1}{\mathcal{D}_1 + \mathcal{D}_2} n^{4/7} + \frac{\mathcal{C}_2}{\mathcal{D}_1 + \mathcal{D}_2} n^{-3/7}.$$

For large n, the approximations should be good. Figure 3.2 shows two examples of the relative root mean squared error, first test distribution 2 with $m = 2$, and then test distribution 3 for $m = 4$. The first plot shows the asymptotic bias effect quite well. The second needs $n > 1000$ for the bias effect to be visible.

Figure 3.3 shows the same for test distribution 4. In this case the kernel estimate is the mean since $g'''(y_0) = 0$. From table A.2 we know that the least false estimate has a bias of -0.2619 for $m = 1$, while it is 0 for $m = 2$, which explains the increase in performance.

Note that the asymptotics not necessarily hold for low n, so the figures are started at $n = 50$. The discussion on which n is required for the results to be valid is beyond the scope of this thesis. Note that in order to estimate parameters for f_m, n has to me at least $m + 2$ in order to have enough data points for the parameters.

3.3 Simulations

For FIC, the focus parameter is the mode

$$\hat{\mu} = \mu(\hat{\xi}, \hat{\sigma}, \hat{\boldsymbol{a}}) = \arg\max_y \{ f_m(y; \hat{\xi}, \hat{\sigma}, \hat{\boldsymbol{a}}) \}.$$

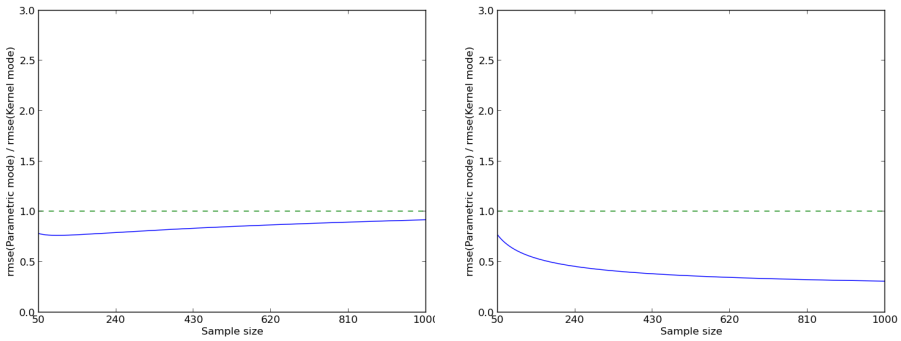


Figure 3.2: Relative root mean squared error for the mode estimate of the parametric model and the kernel estimate, for various sample sizes. To the left for $m = 2$ and test distribution 2, and to the right for $m = 4$ and test distribution 3.

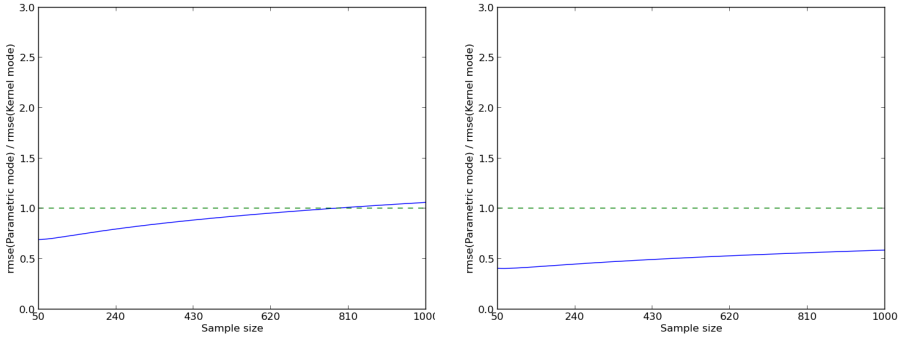


Figure 3.3: Relative root mean squared error for the mode estimate of the parametric model and the kernel estimate, for various sample sizes. To the left for $m = 1$ and test distribution 4, and to the right for $m = 3$ and test distribution 4.

Simulations should give a good impression of the performance of the scheme in chapter 2, since it incorporates both model selection uncertainty, and measures ability to pick the largest mode, and not just any bump.

The main idea behind the simulations is to use Monte Carlo integration to find

$$\text{mse} = E[(\hat{y}_0 - y_0)^2],$$

for both kernel estimates and estimates by the scheme, and then compare the results.

3.3.1 Test Procedure

The tests are conducted as follows:

1. Draw n random variables from (3.1)
2. Estimate FIC and $\hat{\mu}$ for the candidate models, and keep the mode estimate from the model with lowest FIC value.
3. Repeat 1000 times, and denote the i'th estimated mode as $\hat{y}_{0,i}$

For the experiment, we let $m = 5$ be the widest esimating model. The six test distributions in 3.1 are tested, with sample sizes of 50, 200 and 1000, to get an impression of FIC for both varying complexity, and sample size. In other words, a total of 18 tests will be performed in each experiment. For this chapter we will do

1. Regular mode hunt with FIC for candidate models $m = \{0, 1, 2, 3, 4, 5\}$. Both with wide and narrow estimates for J and ω.
2. Regular mode hunt with FIC for candidate models $m = \{2, 3, 4, 5\}$. Both with wide and narrow estimates for J and ω.

For the parametric estimates, the result will be calculated as an estimated mean squared error

$$\widehat{\text{mse}}(\hat{y}_{0,P}) = \sum_{i=1}^{1000} (\hat{y}_{0,i} - y_0)^2.$$

For the kernel estimates, analog simulations will be done with the bandwidth in (1.15). The final result is presented in the table as

$$\text{result in table} = \frac{\widehat{\text{rmse}}(\hat{y}_{0,P})}{\widehat{\text{rmse}}(\hat{y}_{0,K})} = \sqrt{\frac{\widehat{\text{mse}}(\hat{y}_{0,P})}{\widehat{\text{mse}}(\hat{y}_{0,K})}},$$

for each of the 18 test cases. The test distributions range from standard normal, to very non normal. The first is a standard normal distribution, the basis for both model classes. The second and third are quite skewed, while the fourth is completely symmetric. The two last are very similar, with two bumps at similar heights.

3.3.2 Results

The results for the tests are in table 3.3 and 3.4. The tests shows that the wide estimators are better than the narrow estimators. This is not too surprising, since we are outside the locally misspecified framework of the FIC, where the non-narrow parameters goes to zero.

It seems that for the multimodal distributions, it is better to leave $m = 0$ and $m = 1$ out from the model candidates. For test distribution 1, the standard normal distribution the results got worse. This is natural since even though the bias is zero, the variance increases for higher m.

3.4 Summary

In terms of efficiency, the variance of the parametric parameter estimates are $\mathcal{O}(n^{-1})$, while the kernel estimates are $\mathcal{O}(n^{-4/7})$, which gives the parametric estimates an asymptotic advantage. However, the kernel estimates are asymptotically unbiased, but the parametric may be biased.

The results shows that going from $m = \{0, \ldots, 5\}$ to $m = \{2, \ldots, 5\}$ increases performance for the multimodal models, but decreases preformance for test distribution 1. However, the model selection uncertainty induced bias and variance, even though the parametric estimate is better for some m and n.

3.4.1 What model is selected?

Alongside the simulations, the number of times each model was chosen was recorded. These table show the observed probability distribution of which model the FIC scheme choose.

An important question is how biased the mode is. For test distribution 2, 3, 5 and 6, the narrow model is heavily biased, but for 1 and 4 it is not. When it comes to estimating the mode, the narrow estimated FIC was not satisfactory. It turns out that this is because FIC chooses the model with $m = 0$ most of the times. Consider test distribution 3. The observed probability distribution $\hat{\pi}$ for the narrow estimated FIC is

	$m = 0$	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
50	0.5620	0.1290	0.0000	0.1950	0.0000	0.1140
200	0.6210	0.1330	0.0000	0.1670	0.0000	0.0790
1000	0.5500	0.1760	0.0000	0.1450	0.0000	0.1290

which shows that in more than half of the simulations, the narrow model was selected. The narrow model has expected mode 0.6, while the true mode is 0.0. This explains why the FIC did not perform well compared to the kernel. The same table for the wide estimated FIC is

Narrow	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
n=50	1.0951	1.9848	0.7548	1.1429	1.1507	1.0236
n=200	1.4177	4.4800	0.8485	1.7616	3.3176	1.3339
n=1000	1.5232	5.4907	3.2852	3.0181	8.2226	4.6581

(a) Simulation results for mode hunting with narrow estimated FIC, presented as the ratio $\widehat{\text{rmse}}[\hat{y}_{0,P}]/\widehat{\text{rmse}}[\hat{y}_{0,K}]$ from $n = 1000$ simulations. The estimating model is selected from six models from (2.3), with $m = \{0,\ldots,5\}$. The data are generated from (3.1) with parameters from table 3.1

Wide	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
n=50	0.9922	2.3155	0.7599	1.1177	1.2547	0.9562
n=200	1.3556	4.4809	0.6600	1.4568	4.0098	1.4104
n=1000	1.7519	4.3234	0.9394	2.5373	10.6618	3.2008

(b) Simulation results for mode hunting with wide estimated FIC, presented as the ratio $\widehat{\text{rmse}}[\hat{y}_{0,P}]/\widehat{\text{rmse}}[\hat{y}_{0,K}]$ from $n = 1000$ simulations. The estimating model is selected from six models from (2.3), with $m = \{0,\ldots,5\}$. The data are generated from (3.1) with parameters from table 3.1.

Table 3.3

Narrow	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
n=50	1.8036	1.0793	0.8255	0.7746	1.0296	1.0340
n=200	2.0780	1.1380	0.7290	0.7851	0.9810	1.0248
n=1000	2.0114	1.4473	0.9483	1.0271	1.0049	1.5230

(a) Simulation results for mode hunting with narrow estimated FIC, presented as the ratio $\widehat{\text{rmse}}[\hat{y}_{0,P}]/\widehat{\text{rmse}}[\hat{y}_{0,K}]$ from $n = 1000$ simulations. The estimating model is selected from four models from (2.3), with $m = \{2,\ldots,5\}$. The data are generated from (3.1) with parameters from table 3.1

Wide	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
n=50	1.9006	1.1644	1.0400	1.2308	1.0387	1.0114
n=200	2.2835	1.0556	0.7263	1.7206	0.9733	1.0269
n=1000	2.1017	1.4913	0.5126	1.5548	0.7491	0.5371

(b) Simulation results for mode hunting with wide estimated FIC, presented as the ratio $\widehat{\text{rmse}}[\hat{y}_{0,P}]/\widehat{\text{rmse}}[\hat{y}_{0,K}]$ from $n = 1000$ simulations. The estimating model is selected from four models from (2.3), with $m = \{2,\ldots,5\}$. The data are generated from (3.1) with parameters from table 3.1

Table 3.4

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
50	0.3800	0.2380	0.0740	0.0240	0.2730	0.0110
200	0.0230	0.3110	0.0190	0.0330	0.6100	0.0040
1000	0.0000	0.0430	0.0010	0.0030	0.9500	0.0030

which shows that the wide estimators more ofte choose the wider models, which in this case is good. The same two tables for test distribution 6 gave

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
50	0.3940	0.1250	0.0000	0.3400	0.0000	0.1410
200	0.4520	0.1240	0.0000	0.3250	0.0000	0.0990
1000	0.4700	0.1390	0.0000	0.3450	0.0000	0.0460

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
50	0.8740	0.0840	0.0290	0.0040	0.0060	0.0030
200	0.8600	0.0430	0.0820	0.0050	0.0090	0.0010
1000	0.2810	0.0010	0.3660	0.0020	0.3320	0.0180

This shows the same tendency, that the wide estimators has a better turnover from narrow to wide when the sample size grows. For the narrow estimates, the distributions barely change at all.

Chapter 4

Density Estimation with Average-FIC

In this chapter we investigate the modehunter scheme's ability to do density estimation. For reference, we will use the same six test distributions as in the previous chapter, with the mean integrated squared error, defined as

$$\begin{aligned}\text{mise}(\hat{f}, g) &= E\left[\int_\Omega (\hat{f}(y) - g(y))^2 dy\right] \\ &= \int_\Omega (E[\hat{f}(y)] - g(y))^2 dy + \int_\Omega \text{var}(\hat{f}(y)) dy \\ &= \int_\Omega \text{bias}_g(\hat{f}(y)) dy + \int_\Omega \text{var}_g(\hat{f}(y)) dy\end{aligned}$$

as the measure of error.

4.1 Least False Computations

The least false parameters has been found in the previous chapter, and are presented in table A.2 and A.2.

4.1.1 Approximations

We have that with Gaussian kernel, the mise of the kernel estimate of a densty g is approximately

$$\text{mise}(\hat{f}_{n,K}, g) = \frac{1}{4}h^4 \int_{-\infty}^{\infty} g''(y)^2 dy + \frac{1}{2\sqrt{\pi}nh},$$

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
1	0.0049	0.0077	0.0122	0.0340	0.0218	0.0390	0.0786	0.0465	0.0659
2	0.0478	0.0420	0.0131	0.0131	0.0176	0.0178	0.0240	0.0261	0.0542
3	0.0274	0.0287	0.0346	0.0294	0.0138	0.0146	0.0187	0.0191	0.0259
4	0.0577	0.0561	0.0442	0.0347	0.0205	0.0210	0.0239	0.0243	0.0289
5	0.0424	0.0411	0.0121	0.0124	0.0166	0.0182	0.0283	0.0237	0.1744
6	0.0412	0.0420	0.0116	0.0126	0.0184	0.0210	0.0320	0.0247	0.0545

Table 4.1: Approximate mean integrated squared error for distributions f_m (2.3), when estimating test distributions from (3.1) with parameters from table 3.1, for different values of m, and $n = 50$.

which can be computed with numerical methods when we know g. For the parametric estimate, using the least false parameters, we can compute the approximate mean integrated squared. The integrated bias is

$$\int_{-\infty}^{\infty} \text{bias}_g(\hat{f}_m(y))dy = \int_{-\infty}^{\infty} (\hat{f}_{m,P}(y;\xi_0,\sigma_0,\boldsymbol{a}_0) - g(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p}))^2 dy.$$

We know by maximum likelihood theory that $\hat{f}_m(y)$ has an approximate distribution

$$\sqrt{n}(\hat{f}_m(y) - f(y)) \to_D \mathcal{N}\left(0, \frac{\partial f_m(y)}{\partial \zeta_0}^t J(\zeta_0)^{-1} K(\zeta_0) J(\zeta_0)^{-1} \frac{\partial f_m(y)}{\partial \zeta_0}\right),$$

where $\zeta_0 = (\xi_0, \sigma_0, \boldsymbol{a}_0)$, the least false parameters of f when estimating g, with a given set of parameters. This gives that the approximated integrated variance is

$$\int_{-\infty}^{\infty} \text{var}_g(\hat{f}(y))dy = \frac{1}{n}\int_{-\infty}^{\infty} \frac{\partial f_m(y)}{\partial \boldsymbol{\zeta}_0}^t J(\boldsymbol{\zeta}_0)^{-1} K(\boldsymbol{\zeta}_0) J(\boldsymbol{\zeta}_0)^{-1} \frac{\partial f_m(y)}{\partial \boldsymbol{\zeta}_0} dy,$$

which is computed numerically. Examples of approximate mean integrated squared errors are in table 4.1 and 4.2.

4.1.2 Asymptotic Effects and Results

The numbers presented in tables 4.1 and 4.2, tells that there will be cases where the kernel estimate is better, and other cases where the parametric estimate will be better. Just as with the mode, the approximate relative mean integrated squared error can be presented as

$$\frac{\mathcal{C}_1 + \mathcal{C}_2 n^{-1}}{\mathcal{D}_1 n^{-4/5} + \mathcal{D}_2 n^{-4/5}} = \frac{\mathcal{C}_1}{\mathcal{D}_1 + \mathcal{D}_2} n^{4/5} + \frac{\mathcal{C}_2}{\mathcal{D}_1 + \mathcal{D}_2} n^{-1/5}$$

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
50	0.0152	0.0372	0.1253	0.0292	0.0288	0.0292
200	0.0050	0.0123	0.0413	0.0096	0.0095	0.0096
1000	0.0014	0.0034	0.0114	0.0027	0.0026	0.0027

Table 4.2: Approximate mean integrated squared error for the kernel estimate with Silvermans rule-of-thumb bandwidth, when estimating the different test distributions from (3.1) with parameters from table 3.1.

which for $C_1 > 0$ goes to infinity as $n \to \infty$. However, for $C_1 = 0$ it will go towards zero at a speed of $n^{-1/5}$.

In figure 4.1, this effect is shown for test distribution 2 and $m = 4$, and test distribution 3 with $m = 3$. The plots shows the asymptotic bias effect, that the asymptotically unbiased kernel estimate is superior for large sample size. However, for moderate n parametric estimates are still better. Figure 4.2 shows calculations with test distribution 1, the normal distribution, where the experiment has no estimation bias. Both plots shows that the relative mise goes to zero as predicted, in favor of the parametric estimates. However, for $m = 4$, the variance of the parametric estimate is quite high because of the four superfluous parameters.

Note that the asymptotics not necessarily hold for low n, so the figures are started at $n = 50$. The discussion on which n is required for the results to be valid is beyond the scope of this thesis. Note that in order to estimate parameters for f_m, n has to me at least $m + 2$ in order to have enough data points for the parameters.

4.2 Simulations

The tests are very similar to those in the previous chapter. The only difference is that we will not use the narrow estimates for ω and J, because of the less good performance in the mode hunt. The tests are conducted as follows:

1. Draw n random variables from (3.1)
2. Estimate Average-FIC for the candidate models, and record the integrated squared error of the model with lowest AFIC value.
3. Repeat 1000 times, and denote the i'th estimated density as $f_{m,i}$

For the experiment, we let $m = 5$ be the widest esimating model. The six test distributions in 3.1 are tested, with sample sizes of 50, 200 and 1000, to get an impression of Average-FIC for both varying complexity, and sample size. In

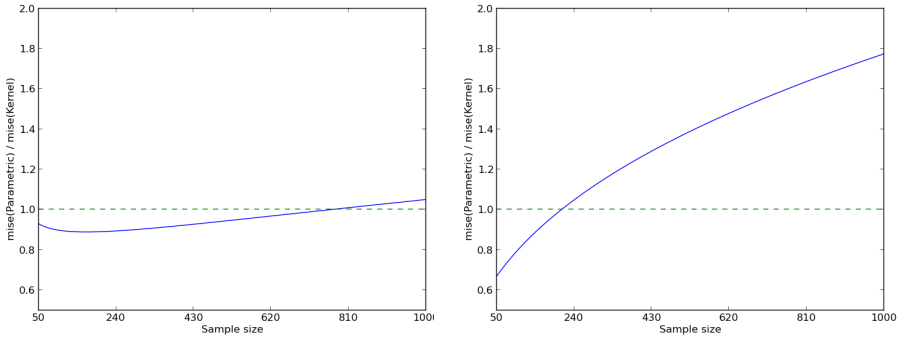


Figure 4.1: Relative mean integrated squared error for probability density estimate of the parametric model and the kernel estimate. The estimates are for the density function of test distribution 2 with $m = 4$ (left) and test distribution 3 with $m = 3$ (right).

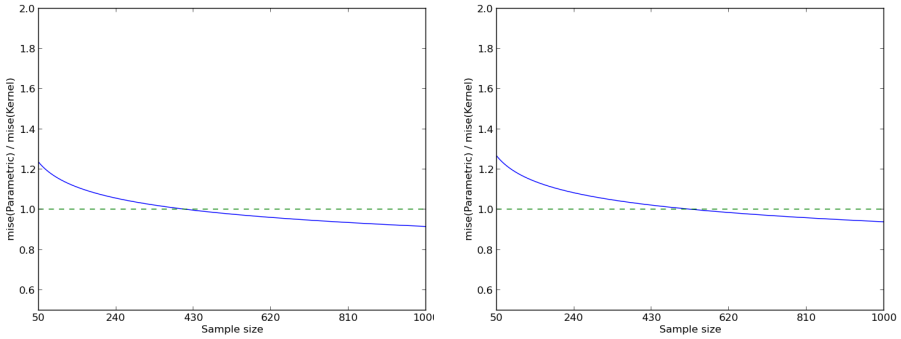


Figure 4.2: Relative Mean Integrated Squared Error for probability density estimate of the parametric model and the kernel estimate. The estimates are for the density function of test distribution 1 with $m = 3$ (left) and model 1 with $m = 4$ (right).

other words, a total of 18 tests will be performed in each experiment. For this chapter we will do

1. Density estimation with AFIC and candidate models $m = \{0, 1, 2, 3, 4, 5\}$, with wide estimates for J and ω.
2. Density estimation with AFIC and candidate models $m = \{2, 3, 4, 5\}$, with wide estimates for J and ω.

For the density estimates, the result will be calculated as an estimated integrated mean squared error, which in this case is computed as

$$\widehat{\text{mise}}[\hat{f}_m] = \frac{1}{n}\sum_{i=1}^{1000}\int_{-10}^{10}(\hat{f}_{(m,i)} - g(y;\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{p}))^2 dy.$$

Completely analog simulations are done with the kernel estimate, with Silverman's rule-of-thumb described in (1.14). The final result is presented in the table as

$$\text{result in table} = \frac{\widehat{\text{mise}}[\hat{f}_m]}{\widehat{\text{mise}}[\hat{f}_{n,K}]}.$$

4.2.1 Results

The results for the tests are in table 4.3 and 4.4. The results shows good performance for the Average-FIC scheme.

4.3 Summary

In terms of efficiency, the variance of the parametric parameter estimates are $\mathcal{O}(n^{-1})$, while the kernel estimates are $\mathcal{O}(n^{-4/5})$, which gives the parametric estimates an asymptotic advantage. However, the kernel estimates are asymptotically unbiased, but the parametric may be biased.

The results shows good performance for the Average-FIC scheme. Also the results shows that going from $m = \{0, \ldots, 5\}$ to $m = \{2, \ldots, 5\}$ increases performance for the multimodal models, but decreases preformance for test distribution 1.

4.3.1 Which model is selected?

Alongside the simulations, the number of times each model was chosen was recorded. These tables shows the observed probability distribution of which model the Average-FIC scheme choose.

The first test, with $m = \{0, \ldots, 5\}$ shows that Average-FIC succeeds in choosing $m = 0$ for test distribution 1. This is good since both is the standard normal distribution.

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
50	0.6710	0.0890	0.1280	0.0290	0.0670	0.0160
200	0.7910	0.0730	0.0700	0.0180	0.0410	0.0070
1000	0.7580	0.0970	0.1030	0.0120	0.0250	0.0050

Also for higher n, it picks $m = 0$ most of the times. For test distribution 5 the Average-FIC shows a good progression from narrower to wider, as n increases

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
50	0.0630	0.0250	0.6110	0.1400	0.1530	0.0080
200	0.0020	0.0010	0.6400	0.1030	0.2400	0.0140
1000	0.0000	0.0000	0.2990	0.0300	0.6420	0.0290

.

For the second test, with candidate models $m = \{2, \ldots, 5\}$, there are similar patterns. For test distribution 1, the standard normal distribution, we have

	$m=2$	$m=3$	$m=4$	$m=5$
50	0.6880	0.1690	0.1290	0.0140
200	0.7520	0.1490	0.0860	0.0130
1000	0.8140	0.1070	0.0730	0.0060

’

which shows that FIC is good at choosing the model closest to the test distribution. For model 5 we have

	$m=2$	$m=3$	$m=4$	$m=5$
50	0.6220	0.1840	0.1790	0.0150
200	0.6690	0.1150	0.2060	0.0100
1000	0.3220	0.0420	0.6130	0.0230

which also shows progression from narrow to wide as n grows.

Wide	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
n=50	1.0457	1.4171	1.3153	1.9922	1.2169	1.2183
n=200	0.5875	1.0030	1.0829	1.6276	0.8067	0.7626
n=1000	0.3877	0.9379	0.8759	2.8474	0.7830	0.7639

Table 4.3: Simulation results for mode hunting with Average-FIC, presented as the ratio $\hat{\text{mise}}[\hat{f}_m]/\hat{\text{mise}}[\hat{f}_{n,K}]$ from $n = 1000$ simulations. The estimating model is selected from six models from (2.3), with $m = \{0, \ldots, 5\}$. The data are generated from (3.1) with parameters from table 3.1.

Wide	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
n=50	1.9148	1.0521	0.9299	1.2909	1.0887	1.1282
n=200	1.2062	0.8114	0.4070	1.1496	0.7852	0.7423
n=1000	0.7615	1.0211	0.3925	2.8377	0.7860	0.7704

Table 4.4: Simulation results for mode hunting with Average-FIC, presented as the ratio $\hat{\text{mise}}[\hat{f}_m]/\hat{\text{mise}}[\hat{f}_{n,K}]$ from $n = 1000$ simulations. The estimating model is selected from four models from (2.3), with $m = \{2, \ldots, 5\}$. The data are generated from (3.1) with parameters from table 3.1.

Chapter 5

Conclusions and Outlook

5.1 Conclusions

There are several results in this thesis that are noteworthy in the conclusion, however the most important ones are

1. Variance and bias related to that an information criterion might pick the wrong model
2. Optimality in kernel estimates of the mode
3. The asymptotic bias effect
4. 'Active' model selection

5.1.1 Model Selection Uncertainty - Wide vs. Narrow

This concept was mentioned in section 1.3, and depicted through observed outcomes from the FIC simulations in chapter 3. In the mode hunt, the narrow estimated FIC turned out to be very conservative, and picking the narrow model most of the times, even for large datasets that were far from normal. This improved when going to wide estimated FIC.

The effect from narrow to wide was bigger than anticipated, however not surprising. The wide estimated FIC should be more robust to misspecification, and turned out to be so in both the mode hunt and the density estimation experiments.

5.1.2 Optimality in Kernel Mode Estimation

The mathematics leading to the optimal kernel bandwidth, is an adaption from similar derivations in Eddy (1980). The estimation involved in deciding the optimal bandwidth are quite demanding, however when they are established they turn out to perform well.

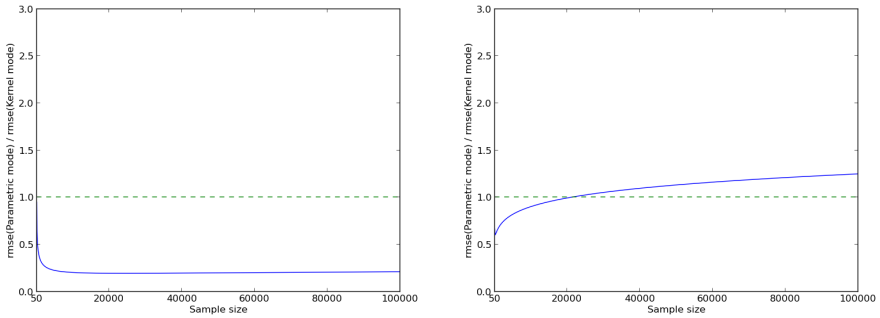


Figure 5.1: Relative root mean squared error for the mode estimate of the parametric model, for different sample size. To the left $m = 5$ and test distribution 5, to the right $m = 5$ and test distribution 2.

5.1.3 The Asymptotic Bias Effect

The asymptotic bias effect is something one should have in mind when estimating focus parameters, with FIC or any other model selection scheme. When the sample size is sufficiently large, one should rely on asymptotically unbiased estimation methods, rather than taking the risk of guessing on the true data generating family of distributions.

The effect is visible in both the mode hunt, and the density estimation experiments. It shows clearly that at some point the bias of the parametric estimate dominates the relative mean squared error of the two. The big problem with this observation is that it is near impossible to tell how large the sample size should be before one turns to non-parametric estimation. For example estimating test distribution 5 with $m = 5$, has very little bias, so the asymptotic bias effect is neglectible for any sensible sample size.

Figure 5.1 shows examples where the sample size has to be very large for the bias effect to count.

5.1.4 Active Model Selection

One stragety in model selection, is to throw in a lot of different models to a model selection scheme, and choose the model with the lowest *IC-value[1], without caring much for what models you picked as candidates.

However, including models that are far from the generating distribution g, means that there is a positive probability for the model selector to choose a

[1] *IC means any information criterion, for example AIC, BIC, DIC, TIC, FIC, AFIC, GIC etc. See Claeskens and Hjort (2008) for a more comprehensive list.

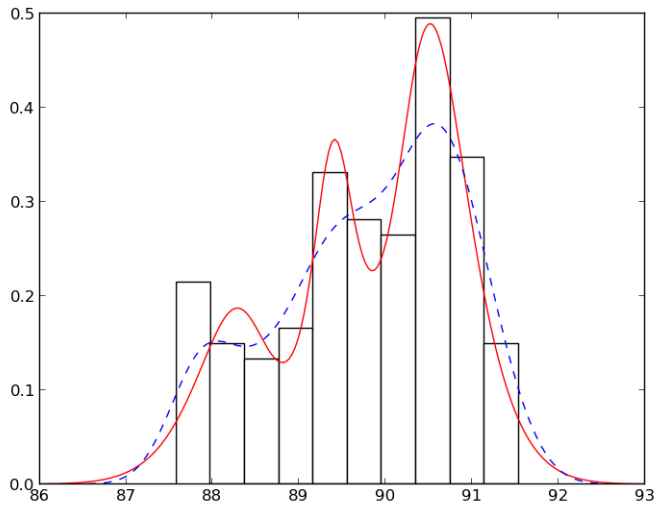


Figure 5.2: Histogram of the Tour de France data with fitted model (2.3) with $m = 4$ (—), and kernel density estimate (- -).

very bad model. This is seen in the FIC and Average-FIC analysis of this thesis, where the standard normal distribution is included, even though the datasets are clearly multimodal. In other words, one should have good arguments for every candidate model included in the analysis.

5.2 Tour de France Overall Times

The Tour de France, or simply The Tour, is an annual multistage bicycle race, held primarily in France. It is one of the most famous of all bicycle races, and gets enormous attention each year. In summer 2012 Bradley Wiggins from UK won with an overall time 87 hours, 34 minutes and 47 seconds. At the competition website www.letour.com the finish times of all the 153 contestants that finished the race are published[2].

5.2.1 Mode Estimation

In context of the thesis, we are interested in estimating the mode of the underlying distribution. In other words the overall times that is the most probable. Figure 5.2 shows that the data has potential bumps, but it is not very far from

[2]http://www.letour.com/le-tour/2012/us/stage-20/classifications.html

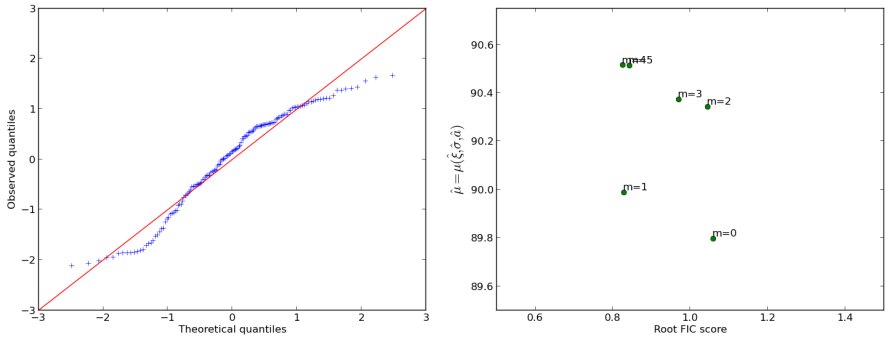


Figure 5.3: Q-Q Plot of the Tour de France dataset to check for normality (left), and FIC plot with FIC scores on the x-axis and focus parameter estimates on the y-axis for all model candidates (right).

normal either, which makes the analysis difficult. A regular wide FIC analysis with model candidates $m = \{0, \ldots, 5\}$ gave the table

Model	st.dev	bias	rFIC	$\hat{\mu}$
m = 0	0.7309	0.7665	1.0591	89.7951
m = 1	0.7794	0.2833	0.8293	89.9873
m = 2	0.7828	0.6928	1.0454	90.3415
m = 3	0.8076	0.5384	0.9706	90.3721
m = 4	0.8257	-0.1715	0.8258*	90.5134
m = 5	0.8435	0.0000	0.8435	90.5128

which indicates multimodality. The FIC mode also couples quite well with the kernel mode, which was estimated to be 90.4657. The big question in this example is whether the dataset is multimodal or not. If it is not, we probably have a very biased estimate. The Q-Q plot in figure 5.3 shows that the data are quite close to normal, but not quite.

Since the chosen model was larger than 2, removing the two narrower models from the analysis will give the same result.

5.2.2 Density Estimation

Average-FIC picked the same model for density estimation as FIC did for the mode. The Average-FIC table is

Model	st.dev	bias	rAFIC
m=0	0.0000	7.2549	7.2549
m=1	1.1501	5.9794	6.0890
m=2	1.5097	3.6640	3.9628
m=3	1.8650	3.6528	4.1013
m=4	2.1430	0.0000	2.1430
m=5	2.1814	0.0000	2.1814

So for this analysis, we will conclude that the dataset is multimodal.

5.3 Ideas for Future Work

5.3.1 Improving the FIC

The FIC has some weaknesses, and the most apparent is that it is a first order Taylor approximation. In chapter 1 we show that

$$f\left(y;\boldsymbol{\theta},\gamma_0+\frac{\delta}{\sqrt{n}}\right)=f\left(y;\boldsymbol{\theta},\gamma_0+\frac{\delta}{\sqrt{n}}\right)\left(1+\frac{\delta}{\sqrt{n}}\frac{\partial\ell}{\partial\gamma_0}\right)+o(h) \tag{5.1}$$

which is a first order approximation. This is later combined with the Δ-method which is also a first order approximation. The ω might be complicated, and the density f may be far from linear when γ is not close to γ_0. This combined will perhaps give very unsecure estimates of the bias.

5.3.2 Estimating other foci

The use of AFIC to do density estimation has many applications, if proven successful. Some examples of possible focusses are

$$\mu(\boldsymbol{\theta},\gamma)=\int_{-\infty}^{\infty}g(y)\log g(y)dy$$

which can be used to calculate a more exact Kullback-Leibler distance, and even calculate exact AIC or TIC values. This is useful in order to tell how good the model is, instead of just comparing different models to each other. Another focus parameter is

$$\mu(\boldsymbol{\theta},\boldsymbol{\gamma})=\int_{-\infty}^{\infty}[g''(y)]^2dy$$

which can be used to determine the optimal bandwidth in kernel density estimation, in equation (1.12). Also if one wish to use L_2 norm based model fit and information criterions, the focus parameter

$$\mu(\boldsymbol{\theta},\boldsymbol{\gamma})=\int_{-\infty}^{\infty}[g(y)]^2dy$$

will be useful. In for instance actuarial science, the points where it is 99% chance of going bankrupt is the standard for how safe the funding should be, in that case we have

$$\mu(\boldsymbol{\theta}, \boldsymbol{\gamma}) = G^{-1}(u).$$

Other percentiles could be interesting as well, for example the median, where $u = \frac{1}{2}$.

5.3.3 Multidimensional Density Estimation

In Silverman (1986), the multi dimensional analogues to kernel estimation are discussed. A multi dimensional variant of (2.3) should be possible. For two dimensions one possibility is

$$f_{m,n}(\boldsymbol{y}; \boldsymbol{\xi}, A, \boldsymbol{a}, \boldsymbol{b}) = \phi_2(\boldsymbol{y}; \boldsymbol{\xi}, A) \frac{1}{\sigma} \exp\left(\sum_{j=1}^{m} a_j \psi_j(\Phi(u_1)) + \sum_{j=1}^{n} b_j \psi_j(\Phi(u_2)) \right) \frac{1}{k_m(\boldsymbol{a}, \boldsymbol{b})}$$

where ϕ_2 is the multinormal distribution in dimension 2, Φ is the normal cumulative function in dimension 1, and

$$u_1 = \frac{y_1 - \xi_1}{A_{0,0}}, \qquad u_2 = \frac{y_2 - \xi_2}{A_{1,1}},$$

and A is the covariance matrix of $\boldsymbol{y}$. Examples of usage, allthough this probably should be modelled with a space-time model, is the tracking of Michael Jackson's white glove in a national television broadcasted live version of 'Billie Jean'[3].

See figure 5.4 for scatterplot, and a two dimensional kernel estimate with bivariate normal kernel function.

5.3.4 Model Averaging

Assume that we have done FIC analysis on some focus parameter μ, then the final estimate can be represented as

$$\hat{\mu}_{fic}(S) = \sum_{S \in \mathcal{A}} w(S) \hat{\mu}(S)$$

where $w(S) = I(S = S_{fic})$, the indicator of model S having the smallest FIC score. The natural extension is to use different weight functions. Claeskens and Hjort (2008) suggests using weights

$$w(S) = \frac{\exp\left(-\frac{\kappa FIC(S)}{2\hat{\omega}^t \hat{Q} \hat{\omega}}\right)}{\sum_{\text{all } S} \exp\left(-\frac{\kappa FIC(S)}{2\hat{\omega}^t \hat{Q} \hat{\omega}}\right)}$$

[3] See http://www.whiteglovetracking.com/

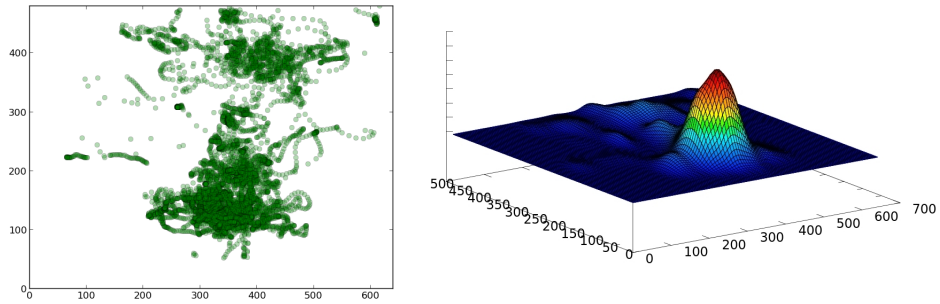


Figure 5.4: Tracking Michael Jackson's white glove. Scatterplot (left) and Kernel Density Estimate (right).

to average over the different $\hat{\mu}_S$. The parameter κ is a smoothing parameter, which takes the weights from uniform ($\kappa = 0$), to the indicator function above ($\kappa = \infty$).

5.3.5 Tests for Multimodality

One problem with both model selection, and bandwidth selection, is the different properties of unimodal and multimodal distributions. In Cox (1966), a method to identify non-linear parts of the true density is proposed using histograms. Construct a histogram, and let n_i be the number of observations i bin i. If the true probabilities of an observation to en up in bin $i-1$, i and $i+1$ has an approximate linear relation, then $n_i|n_{i-1}, n_i, n_{i+1}$ has an approximate binomial $b(n_{i-1}+n_i+n_{i+1}, \frac{1}{3})$-distribution. Hence the test observer

$$t_i = \frac{n_{i+1} - 2n_i + n_{i-1}}{\sqrt{2(n_{i-1}+n_i+n_{i+1})}}$$

can be shown, by a normal approximation of the binomial distribution, to have a standard normal distribuion. Significant values of t_i will indicate concavity or convexity of the true distribuion.

5.3.6 Other Base Distributions for f_m

In the introduction of chapter 2, a general form of the log expanded models is presented, namely

$$f_m(y;\boldsymbol{a}) = f_0 \cdot \exp\left(\sum_{k=1}^{m} a_k \psi_k(F_0(y))\right) \frac{1}{k_m(\boldsymbol{a})}, \quad k_m(a) = \int_0^1 \exp\left(\sum_{k=1}^{m} a_k \psi_k(y)\right) dy.$$

This opens possibilities for f_0 being almost any continous probability distribution. Consider log expanding a general two parameter member of the exponential class, on the form presented in Knight (2000, p. 188)

$$f_m(y;\boldsymbol{\theta}) = \exp\left[c_1(\theta_1)T_1(y) + c_2(\theta_2)T_2(y) - d(\boldsymbol{\theta}) + S(y) + \sum_{k=1}^{m} a_k\psi_k(F_0(y)) - \log k_m(\boldsymbol{a})\right].$$

This family has a log density function

$$\ell(y;\boldsymbol{\theta}) = c_1(\theta_1)T_1(y) + c_2(\theta_2)T_2(y) - d(\boldsymbol{\theta}) + S(y) + \sum_{k=1}^{m} a_k\psi_k(F_0(y)) - \log k_m(\boldsymbol{a})$$

so we have that at $\boldsymbol{a} = \mathbf{0}$

$$S(y;\boldsymbol{\theta}) = \begin{pmatrix} c_1'(\theta_1)T_1(y) - d'(\theta 1) \\ c_2'(\theta_2)T_2(y) - d'(\theta_2) \\ \psi_1(F_0(y)) \\ \vdots \\ \psi_k(F_0(y)) \end{pmatrix}.$$

For narrow estimated FIC, these models are very neat distributions to work with.

5.3.7 Continue on the Linear Analysis

When using models on the form

$$f_m(y;\boldsymbol{a}) = f_0 \cdot \exp\left(\sum_{k=1}^{m} a_k\psi_k(F_0(y))\right)\frac{1}{k_m(\boldsymbol{a})}$$

as a general scheme for density estimation, it would be interesing to try and see if they are asymptotically consistent. One important suggestion is to see if there is an α, such that

$$m = n^{\alpha}$$

is optimal in terms of mean integrated squared error. The number of parameters m should grow to infinity at some speed related to sample size n. There are some results describing this in Barron and Sheu (1991) in terms of Kullback-Leibler divergence, but they are not general enough for this particular thesis. In order to compare to the kernel estimate, the mise is better.

Appendix A

Tables and Figures

A.1 Tables for the Misspecified Test

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.1580	0.1384	0.0168	0.0139	0.0078	0.0075
3	0.1293	0.1265	0.1258	0.1107	0.0116	0.0115
4	0.2031	0.1986	0.1778	0.1514	0.0349	0.0349
5	0.1426	0.1364	0.0057	0.0057	0.0037	0.0036
6	0.1408	0.1393	0.0056	0.0056	0.0037	0.0037

Table A.1: The least false Kullback-Leibler divergences for different m of class (2.3), when estimating class (3.1) distributions in table 3.1. The computations are done as described in section 3.2.1.

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.8000	1.1450	1.8078	1.8129	1.8542	1.8585
3	0.6000	0.9153	1.1301	-0.3395	-0.0220	-0.0174
4	0.0000	-0.2619	0.0000	-0.1441	0.0000	-0.0003
5	0.8000	1.0464	1.9012	1.8990	1.9741	1.9806
6	0.6500	0.7909	1.9018	1.9004	1.9734	1.9768

Table A.2: The least false mode estimates for different m of class (2.3), when estimating modes of the class (3.1) distributions in table 3.1. The computations are done as described in section 3.2.1.

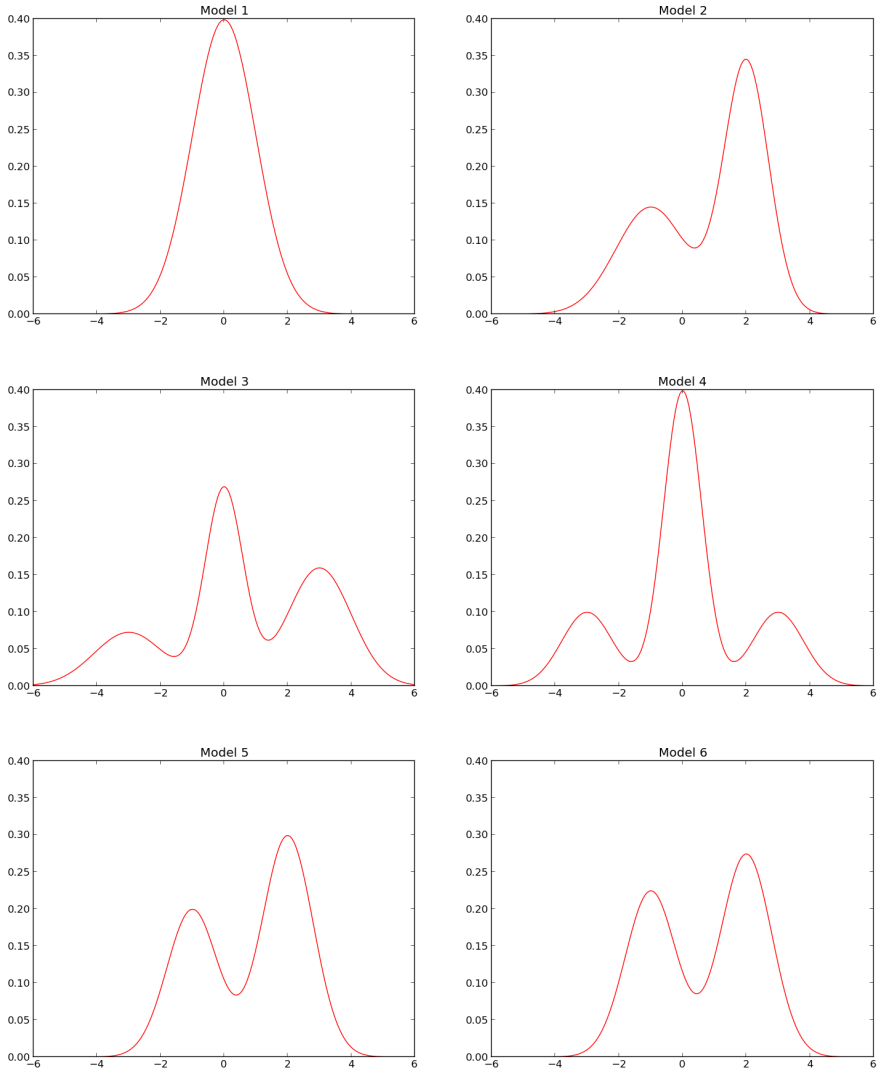


Figure A.1: Plots of the six normal mixture distributions that are used as reference in the tests. The probability density functions are defined in (3.1), while the parameters are in 3.1.

	ξ_0	σ_0	$\boldsymbol{a}_0$	$\hat{D}_{KL}$	Mode
m=0	0.0000	1.0000	()	0.0000	0.0000
m=1	0.0000	1.0000	(0.0000)	0.0000	0.0000
m=2	0.0000	1.0000	(0.0000, 0.0000)	0.0000	0.0000
m=3	0.0000	1.0000	(0.0000, 0.0000, 0.0000)	0.0000	0.0000
m=4	0.0000	1.0000	(0.0000, 0.0000, 0.0000, 0.0000)	0.0000	0.0000
m=5	0.0000	1.0000	(0.0000, 0.0000, 0.0000, 0.0000, 0.0000)	0.0000	0.0000

(a) Test distribution 1. True mode: 0.0000

	ξ_0	σ_0	$\boldsymbol{a}_0$	$\hat{D}_{KL}$	Mode
m=0	0.8000	1.7141	()	0.1580	0.8000
m=1	0.3929	1.7237	(-0.3139)	0.1384	1.1450
m=2	0.3131	1.3366	(-0.2613, 0.7293)	0.0168	1.8078
m=3	0.1619	1.3556	(-0.2414, 0.7115, -0.1695)	0.0139	1.8129
m=4	0.0890	1.3079	(-0.2203, 0.6805, -0.2213, 0.1734)	0.0078	1.8542
m=5	0.0301	1.3146	(-0.2194, 0.6598, -0.2438, 0.1856, -0.0433)	0.0075	1.8585

(b) Test distribution 2. True mode: 2.0000

	ξ_0	σ_0	$\boldsymbol{a}_0$	$\hat{D}_{KL}$	Mode
m=0	0.6000	2.4137	()	0.1293	0.6000
m=1	0.1619	2.4411	(-0.1967)	0.1265	0.9153
m=2	0.3211	2.3668	(-0.1313, 0.0567)	0.1258	1.1301
m=3	0.5639	2.1835	(-0.0208, 0.1411, -0.1932)	0.1107	-0.3395
m=4	-0.0413	2.0096	(-0.2371, 0.1278, -0.0423, 0.5756)	0.0116	-0.0220
m=5	-0.0729	2.0117	(-0.2452, 0.1287, -0.0253, 0.5750, -0.0296)	0.0115	-0.0174

(c) Test distribution 3. True mode: 0.0000

Figure A.2: Parameters that minimize the Kullback-Leibler distance between (2.3) for $m = \{0, \ldots, 5\}$, and class (3.1) with the parameters from table 3.1 distributions 1 to 3.

	ξ_0	σ_0	$\boldsymbol{a}_0$	$\hat{D}_{KL}$	Mode
m=0	0.0000	2.0179	()	0.2031	0.0000
m=1	0.9152	2.1130	(0.4660)	0.1986	-0.2619
m=2	0.0098	2.2415	(0.0059, -0.2708)	0.1778	0.0000
m=3	0.8311	2.0782	(0.4391, 0.0281, -0.3212)	0.1514	-0.1441
m=4	0.0001	1.9654	(0.0003, -0.1289, -0.0001, 0.5921)	0.0349	0.0000
m=5	-0.0029	1.9655	(-0.0017, -0.1289, 0.0023, 0.5920, -0.0018)	0.0349	-0.0003

(a) Test distribution 4. True mode: 0.0000

	ξ_0	σ_0	$\boldsymbol{a}_0$	$\hat{D}_{KL}$	Mode
m=0	0.8000	1.6733	()	0.1426	0.8000
m=1	0.5751	1.6749	(-0.1757)	0.1364	1.0464
m=2	0.4780	1.2722	(-0.1638, 0.7570)	0.0057	1.9012
m=3	0.4950	1.2717	(-0.1668, 0.7563, 0.0196)	0.0057	1.8990
m=4	0.5212	1.2298	(-0.1741, 0.7498, 0.0516, 0.0997)	0.0037	1.9741
m=5	0.5084	1.2301	(-0.1725, 0.7525, 0.0443, 0.0976, -0.0105)	0.0036	1.9806

(b) Test distribution 5. True mode: 2.0000

	ξ_0	σ_0	$\boldsymbol{a}_0$	$\hat{D}_{KL}$	Mode
m=0	0.6500	1.6934	()	0.1408	0.6500
m=1	0.5382	1.6937	(-0.0866)	0.1393	0.7909
m=2	0.4891	1.2710	(-0.0811, 0.7612)	0.0056	1.9018
m=3	0.5021	1.2706	(-0.0833, 0.7610, 0.0147)	0.0056	1.9004
m=4	0.5110	1.2308	(-0.0862, 0.7561, 0.0260, 0.0954)	0.0037	1.9734
m=5	0.5075	1.2312	(-0.0862, 0.7566, 0.0254, 0.0945, -0.0048)	0.0037	1.9768

(c) Test distribution 6. True mode: 2.0000

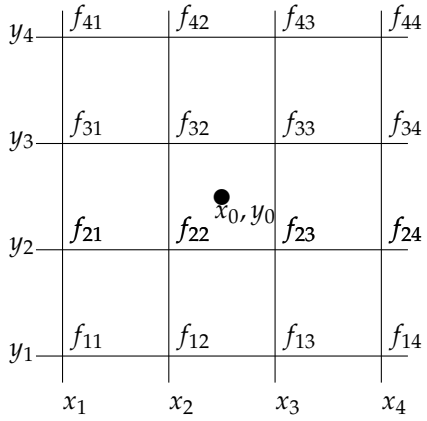
Figure A.3: Parameters that minimize the Kullback-Leibler distance between (2.3) for $m = \{0, \ldots, 5\}$, and class (3.1) with the parameters from table 3.1 distributions 4 to 6.

Appendix B

Numerical Methods and Computations

B.1 A Method for Numerical Hessian Computation

Assume the following grid in the plane, surrounding a point (x_0, y_0), where we want to estimate the mixed partial derivative. The distance between the lines in the grid is h either way.



The standard way of obtaining it, is to approximate the function in the immediate vicinity of (x_0, y_0) with a polynomial, and then differentiate the polynomial. Lagrange interpolation, generalized for tensor product polynomials, is suitable for this. See Dahlquist and Bjorck (2008, p. 395) for details.

The lagrange polynomial of line k along the x-axis is

$$p_k(x) = f_{k1}\frac{x-x_2}{x_1-x_2}\frac{x-x_3}{x_1-x_3}\frac{x-x_4}{x_1-x_4} + f_{k2}\frac{x-x_1}{x_2-x_1}\frac{x-x_3}{x_2-x_3}\frac{x-x_4}{x_2-x_4}$$
$$+ f_{k3}\frac{x-x_1}{x_3-x_1}\frac{x-x_2}{x_3-x_2}\frac{x-x_4}{x_3-x_4} + f_{k4}\frac{x-x_1}{x_4-x_1}\frac{x-x_2}{x_4-x_2}\frac{x-x_3}{x_4-x_3}$$

where

$$(x_1-x_2)(x_1-x_3)(x_1-x_4) = (-h)(-2h)(-3h) = -6h^3$$
$$(x_2-x_1)(x_2-x_3)(x_2-x_4) = (h)(-h)(-2h) = 2h^3$$
$$(x_3-x_1)(x_3-x_2)(x_3-x_4) = (2h)(h)(-h) = -2h^3$$
$$(x_4-x_1)(x_4-x_2)(x_4-x_3) = (3h)(2h)(h) = 6h^3.$$

The next step is to interpolate along the y-axis bewteen the lines interpolated above. In other words, interpolate in y-direction between the polynomials p_k.

$$p(x,y) = p_1(x)\frac{y-y_2}{y_1-y_2}\frac{y-y_3}{y_1-y_3}\frac{y-y_4}{y_1-y_4} + p_2(x)\frac{y-y_1}{y_2-y_1}\frac{y-y_3}{y_2-y_3}\frac{y-y_4}{y_2-y_4}$$
$$+ p_3(x)\frac{y-y_1}{y_3-y_1}\frac{y-y_2}{y_3-y_2}\frac{y-y_4}{y_3-y_4} + p_4(x)\frac{y-y_1}{y_4-y_1}\frac{y-y_2}{y_4-y_2}\frac{y-y_3}{y_4-y_3}$$
$$= p_1(x)q_1(y) + p_2(x)q_2(y) + p_3(x)q_3(y) + p_4(x)q_4(y),$$

where

$$(y_1-y_2)(y_1-y_3)(y_1-y_4) = (-h)(-2h)(-3h) = -6h^3$$
$$(y_2-y_1)(y_2-y_3)(y_2-y_4) = (h)(-h)(-2h) = 2h^3$$
$$(y_3-y_1)(y_3-y_2)(y_3-y_4) = (2h)(h)(-h) = -2h^3$$
$$(y_4-y_1)(y_4-y_2)(y_4-y_3) = (3h)(2h)(h) = 6h^3.$$

Using the triple product rule we have that

$$[(z-z_1)(z-z_2)(z-z_3)]' = (z-z_1)(z-z_2) + (z-z_1)(z-z_3) + (z-z_2)(z-z_3)$$

which leaves that for any k

$$\begin{aligned}\frac{d}{dx}[p_k(x)] = &\frac{-f_{k1}}{6h^3}\left((x-x_2)(x-x_3)+(x-x_2)(x-x_4)+(x-x_3)(x-x_4)\right)\\ &+\frac{f_{k2}}{2h^3}\left((x-x_1)(x-x_3)+(x-x_1)(x-x_4)+(x-x_3)(x-x_4)\right)\\ &+\frac{-f_{k3}}{2h^3}\left((x-x_1)(x-x_2)+(x-x_1)(x-x_4)+(x-x_2)(x-x_4)\right)\\ &+\frac{f_{k4}}{6h^3}\left((x-x_1)(x-x_2)+(x-x_1)(x-x_3)+(x-x_2)(x-x_3)\right),\end{aligned}$$

such that

$$\begin{aligned}p'_k(x_0) = &\frac{-f_{k1}}{6h^3}\left(\frac{h}{2}\left(-\frac{h}{2}\right)+\frac{h}{2}\left(-\frac{3h}{2}\right)+\left(-\frac{h}{2}\right)\left(-\frac{3h}{2}\right)\right)\\ &+\frac{f_{k2}}{2h^3}\left(\frac{3h}{2}\left(-\frac{h}{2}\right)+\frac{3h}{2}\left(-\frac{3h}{2}\right)+\left(-\frac{h}{2}\right)\left(-\frac{3h}{2}\right)\right)\\ &+\frac{-f_{k3}}{2h^3}\left(\frac{3h}{2}\frac{h}{2}+\frac{3h}{2}\left(-\frac{3h}{2}\right)+\frac{h}{2}\left(-\frac{3h}{2}\right)\right)\\ &+\frac{f_{k4}}{6h^3}\left(\frac{3h}{2}\frac{h}{2}+\frac{3h}{2}\left(-\frac{h}{2}\right)+\frac{h}{2}\left(-\frac{h}{2}\right)\right)\\ &=\frac{-f_{k1}}{6h^3}\left(-\frac{1}{4}h^2-\frac{3}{4}h^2+\frac{3}{4}h^2\right)+\frac{f_{k2}}{2h^3}\left(-\frac{3}{4}h^2-\frac{9}{4}h^2+\frac{3}{4}h^2\right)\\ &-\frac{f_{k3}}{2h^3}\left(\frac{3}{4}h^2-\frac{9}{4}h^2-\frac{3}{4}h^2\right)+\frac{f_{k4}}{6h^3}\left(\frac{3}{4}h^2-\frac{3}{4}h^2-\frac{1}{4}h^2\right)\\ &=\frac{1}{8h}\left(\frac{f_{k1}}{3}-9f_{k2}+9f_{k3}-\frac{f_{k4}}{3}\right),\end{aligned}$$

and

$$\begin{aligned}q'_1(y) &= -\frac{1}{6h^3}\left((y-y_2)(y-y_3)+(y-y_2)(y-y_4)+(y-y_3)(y-y_4)\right)\\ q'_2(y) &= \frac{1}{2h^3}\left((y-y_1)(y-y_3)+(y-y_1)(y-y_4)+(y-y_3)(y-y_4)\right)\\ q'_3(y) &= -\frac{1}{2h^3}\left((y-y_1)(y-y_2)+(y-y_1)(y-y_4)+(y-y_2)(y-y_4)\right)\\ q'_4(y) &= \frac{1}{6h^3}\left((y-y_1)(y-y_2)+(y-y_1)(y-y_3)+(y-y_2)(y-y_3)\right),\end{aligned}$$

which gives that

$$q_1'(y_0) = \frac{1}{24h}$$

$$q_2'(y_0) = -\frac{9}{8h}$$

$$q_3'(y_0) = \frac{9}{8h}$$

$$q_4'(y_0) = -\frac{1}{24h}.$$

We are now ready to state the mixed partial derivative of the interpolation polynomial. It is

$$\begin{aligned}\frac{\partial p(x,y)}{\partial x \partial y} &= \frac{1}{8h}\left(\frac{f_{11}}{3} - 9f_{12} + 9f_{13} - \frac{f_{14}}{3}\right)\frac{1}{24h} - \frac{1}{8h}\left(\frac{f_{21}}{3} - 9f_{22} + 9f_{23} - \frac{f_{24}}{3}\right)\frac{9}{8h} \\ &\quad + \frac{1}{8h}\left(\frac{f_{31}}{3} - 9f_{32} + 9f_{33} - \frac{f_{34}}{3}\right)\frac{9}{8h} - \frac{1}{8h}\left(\frac{f_{41}}{3} - 9f_{42} + 9f_{43} - \frac{f_{44}}{3}\right)\frac{1}{24h} \\ &= \begin{pmatrix} \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{pmatrix} \begin{pmatrix} \frac{1}{9} \\ -3 \\ 3 \\ -\frac{1}{9} \end{pmatrix} \frac{1}{8h^2} \qquad \text{(B.1)}\end{aligned}$$

For the one way double derivatives, we get that

$$\begin{aligned}p_k''(x) &= \frac{-f_{11}}{6h^3}\left((x-x_2)+(x-x_3)+(x-x_2)+(x-x_4)+(x-x_3)+(x-x_4)\right) \\ &\quad + \frac{f_{12}}{2h^3}\left((x-x_1)+(x-x_3)+(x-x_1)+(x-x_4)+(x-x_3)+(x-x_4)\right) \\ &\quad + \frac{-f_{13}}{2h^3}\left((x-x_1)+(x-x_2)+(x-x_1)+(x-x_4)+(x-x_2)+(x-x_4)\right) \\ &\quad + \frac{f_{14}}{6h^3}\left((x-x_1)+(x-x_2)+(x-x_1)+(x-x_3)+(x-x_2)+(x-x_3)\right) \\ p''(x_0) &= \frac{-f_{11}}{6h^3}\left(\frac{h}{2}-\frac{h}{2}+\frac{h}{2}-\frac{3h}{2}-\frac{h}{2}-\frac{3h}{2}\right) + \frac{f_{12}}{2h^3}\left(\frac{3h}{2}-\frac{h}{2}+\frac{3h}{2}-\frac{3h}{2}-\frac{h}{2}-\frac{3h}{2}\right) \\ &\quad + \frac{-f_{13}}{2h^3}\left(\frac{3h}{2}+\frac{h}{2}+\frac{3h}{2}-\frac{3h}{2}+\frac{h}{2}-\frac{3h}{2}\right) + \frac{f_{14}}{6h^3}\left(\frac{3h}{2}+\frac{h}{2}+\frac{3h}{2}-\frac{h}{2}+\frac{h}{2}-\frac{h}{2}\right) \\ &= \frac{f_{11}}{6h^3}\frac{6h}{2} - \frac{f_{12}}{2h^3}\frac{2h}{2} - \frac{f_{13}}{2h^3}\frac{2h}{2} + \frac{f_{14}}{6h^3}\frac{6h}{2} = \frac{1}{2h^2}\left(f_{11} - f_{12} - f_{13} + f_{14}\right)\end{aligned}$$

This example program for the function $f(x,y) = \sin(x)\sin(y)$ did the calculations at $(x_0, y_0) = (\frac{1}{2}, \frac{1}{2})$ with an error of $6.7690e-08$ (Euclidian norm).

```
1  from numpy import *
2
3  def mixed(f, x0, y0, h=1e-3):
4      A = matrix(zeros((4, 4)))
5      y1 = matrix([1.0/3, -9.0, 9.0, -1.0/3])
6      y2 = matrix([1.0/24, -9.0/8, 9.0/8, -1.0/24])
7
8      for i in range(4):
9          for j in range(4):
10             A[i, j] = f(x0 - 3.0*h/2 + i*h, y0 - 3.0*h/2 + j*h)
11
12     return y2 * A * y1.T / (8*h**2)
13
14 def double(f, x0, y0, h=1e-3):
15     d1 = zeros(4)
16     d2 = zeros(4)
17     sign = array([1, -1, -1, 1])
18     for i in range(4):
19         d1[i] = f(x0 - 3.0*h/2 + i*h, y0)
20         d2[i] = f(x0, x0 - 3.0*h/2 + i*h)
21     return sum(sign*d1) / (2*h**2), sum(sign*d2) / (2*h**2)
22
23 def hessian(f, x0, y0, h=1e-3):
24     double1, double2 = double(f, x0, y0)
25     d_mixed  = mixed(f, x0, y0)
26
27     return array([[double1, d_mixed], [d_mixed, double2]])
```

B.1.1 Error Analysis for Mixed Derivatives

Error analysis in algorithms is generally divided into two parts; the truncation error and the round-off error Dahlquist and Bjorck (2008, p. 88). The latter occurs when the number system we operate on has a limitation in how many digits that can be stored. For example π has to be rounded off in order to make it fit into the IEEE standard, which is used by Python [1].

[1] http://docs.python.org/2/tutorial/floatingpoint.html

Round-Off Error

Assume that the function values in (B.1) cannot be represented exact on the numeral system used. We get that

$$\begin{aligned}\frac{\partial p(x,y)}{\partial x \partial y} + E_{ro} \leq & \frac{1}{8h}\left(\frac{f_{11}}{9}(1+\epsilon) - 3f_{12}(1+\epsilon) + 3f_{13}(1+\epsilon) - \frac{f_{14}}{9}(1+\epsilon)\right)\frac{1}{8h} \\ & - \frac{1}{8h}\left(\frac{f_{21}}{3}(1+\epsilon) - 9f_{22}(1+\epsilon) + 9f_{23}(1+\epsilon) - \frac{f_{24}}{3}(1+\epsilon)\right)\frac{9}{8h} \\ & + \frac{1}{8h}\left(\frac{f_{31}}{3}(1+\epsilon) - 9f_{32}(1+\epsilon) + 9f_{33}(1+\epsilon) - \frac{f_{34}}{3}(1+\epsilon)\right)\frac{9}{8h} \\ & - \frac{1}{8h}\left(\frac{f_{41}}{9}(1+\epsilon) - 3f_{42}(1+\epsilon) + 3f_{43}(1+\epsilon) - \frac{f_{44}}{9}\right)\frac{1}{8h},\end{aligned}$$

which gives that

$$\begin{aligned}|E_{ro}| &\leq \left|\frac{\epsilon}{64h^2}\left(\frac{f_{11}}{9} - 3f_{12} + 3f_{13} - \frac{f_{14}}{9}\right) - \frac{9\epsilon}{64h^2}\left(\frac{f_{21}}{3} - 9f_{22} + 9f_{23} - \frac{f_{24}}{3}\right)\right| \\ &+ \left|\frac{9\epsilon}{64h^2}\left(\frac{f_{31}}{3} - 9f_{32} + 9f_{33} - \frac{f_{34}}{3}\right) - \frac{\epsilon}{64h^2}\left(\frac{f_{41}}{9} - 3f_{42} + 3f_{43} - \frac{f_{44}}{9}\right)\right| \\ &= \left|\frac{\epsilon}{64h^2}\left(\frac{f_{11}}{9} - 3f_{12} + 3f_{13} - \frac{f_{14}}{9} - 3f_{21} + 81f_{22} - 81f_{23} + 9f_{24}\right)\right| \\ &+ \left|\frac{\epsilon}{64h^2}\left(9f_{31} - 81f_{32} + 81f_{33} - 9f_{34} - \frac{f_{41}}{9} + 3f_{42} - 3f_{43} + \frac{f_{44}}{9}\right)\right| \\ &\leq \frac{\epsilon}{64h^2}\left(366 + \frac{4}{9}\right)f^* \approx 5.7257\frac{\epsilon f^*}{h^2},\end{aligned}$$

where f^* is the maximum value of f among the estimated values, and ϵ is the largest machine epsilon on the system in use.

Truncation Error

From Dahlquist and Bjorck (2008, p. 397) we have that the truncation error of the polynomial estimate, with 4 estimating points in either direction, is

$$\begin{aligned}E_{tr} = & \frac{\partial^4 f(\xi_1, y)}{\partial x^4}\frac{\prod_{i=1}^4 (x - x_i)}{4!} + \frac{\partial^4 f(x, \eta_1)}{\partial x^4}\frac{\prod_{i=1}^4 (y - y_i)}{4!} \\ & - \frac{\partial^8 f(\xi_2, \eta_2)}{\partial x^4 \partial y^4}\frac{\prod_{i=1}^4 (x - x_i)}{4!}\frac{\prod_{i=1}^4 (y - y_i)}{4!}\end{aligned}$$

for some $(\xi_1, \eta_1), (\xi_2, \eta_2)$ inner points of $[x_0 - h, x_0 + h] \times [x_0 - h, x_0 + h]$. Some calculus reveals that

$$\left[\frac{d^2}{dx^2}\prod_{i=1}^{4}(x - x_i)\right]_{x=x_0} = -5h^2$$

which gives that

$$|E_{tr}| \leq \left|5h^2\left(\frac{\partial^4 f(\xi_1, y)}{\partial x^4} + \frac{\partial^4 f(x, \eta_1)}{\partial x^4}\right)\right| + \left|\frac{\partial^8 f(\xi_2, \eta_2)}{\partial x^4 \partial y^4}\frac{h^4}{144}\right|.$$

B.1.2 Error Analysis for One Way Double Derivatives

Round-Off Error

The round-off error occurs when any of the function values calculated cannot be represented on a finite numeral system. We have that

$$\begin{aligned}|p''(x) + E_{ro}| &\leq \frac{1}{2h^2}\left(f_{11}(1+\epsilon) - f_{12}(1+\epsilon) - f_{13}(1+\epsilon) + f_{14}(1+\epsilon)\right) \\ &= \frac{1}{2h^2}\left(f_{11} - f_{12} - f_{13} + f_{14}\right) + \frac{1}{2h^2}\left(f_{11}\epsilon - f_{12}\epsilon - f_{13}\epsilon + f_{14}\epsilon\right),\end{aligned}$$

which means that the round off error is bounded by

$$E_{ro} \leq \frac{\epsilon}{2h^2}\left(f_{11} - f_{12} - f_{13} + f_{14}\right) \leq \frac{2M\epsilon}{h^2}$$

where $M = \max_{[x-3h/2, x+3h/2]}\{f(x)\}$.

Truncation Error

We have from Dahlquist and Bjorck (2008, Thm.4.2.3) that the remainder term for one way interpolation at 4 points in x-direction is

$$f(x) - p(x) = \frac{f^{(4)}(\xi)}{4!}\prod_{i=1}^{4}(x - x_i).$$

for some $\xi \in [x - 3h/2, x + 3h/2]$. The second derivative of the remainder evaluated at x_0 is

$$\left[\frac{d^2}{dx^2}\prod_{i=1}^{4}(x - x_i)\right]_{x=x_0} = -5h^2,$$

which means that

$$|f''(x_0) - p''(x_0)| \leq \frac{5M}{24}h^2,$$

where $M = \max_{[x_0-3h/2, x_0+3h/2]}\{f^{(4)}(x)\}$.

B.1.3 Extension to Triple Differentiation

Continuing with the two times one way differentiated polynomial in the previous section, we differentiate once more to get

$$\begin{aligned} p'''(x) &= \frac{-f_{11}}{6h^3}(6) + \frac{f_{12}}{2h^3}(6) + \frac{-f_{13}}{2h^3}(6) + \frac{f_{14}}{6h^3}(6) \\ &= \frac{1}{h^3}\left(-f_{11} + 3f_{12} - 3f_{13} + f_{14}\right). \end{aligned}$$

Appendix C

Python Scripts and Documentation

C.1 Documentation with Examples

Python as a programming language, is easy to organize in terms of modules and separate scripts with classes and methods. The module 'scripts' created for this thesis contains the following modules:

```
--/scripts/
  |- __init__
  |- FIC
  |- AFIC
  |- logexp
  |- model2
  |- init
  |- hessian
  |- kernel
  |- wfunc
```

where the parts and names of the funcions available from the module is defined in the file __init__. For example, if we in a program wish to use the FIC class, the module logexp which has functions related to the distribution (2.3), and we need the jackknife weight function we start the python program with

```
1 from scripts import fic
2 import scripts.logexp as m1
3 from scripts.wfunc import jackknife
```

Note that the logexp module is named 'm1' locally for convenience. We are now ready to use FIC with it's functionality, if we can provide J_{wide}, $\frac{\partial \mu}{\partial \theta}$, $\frac{\partial \mu}{\partial \gamma}$ and $D_n = \sqrt{n}(\hat{\gamma} - \gamma_0)$. These variables can be obtained in the logexp class. The following program generates 50 random numbers from 'model2' which is the module for (3.1), and does FIC for the mode.

```
from numpy import *
from scripts import fic
import scripts.logexp as m1
import scripts.model2 as m2
from scripts import getparam

# Gets the correct parameters for test model 5 and generates
# random variables.
mu, tau, p = getparam(5)
data, err = m2.rgen(mu, tau, p, 50)

# Fitting the wide and the narrow models.
m = 5
fitwide, ll = m1.mle(data, m)
fitnarr = zeros(m+2)
fitnarr[:2] = array([mean(data), std(data)])

# Calculating Jwide, the partial derivatives of mu, and Dn.
Jwide = m1.Jwide(fitnarr)
dudpar = m1.dmu_mode(fitnarr[0], fitnarr)
ddtheta = array(dudpar[:2]).squeeze()
ddgamma = array(dudpar[2:]).squeeze()
Dn = sqrt(50) * fitwide[2:]

# Using the FIC class to get FIC scores and determine the winner.
ficobj = fic(Jwide, ddtheta, ddgamma, Dn)
ficscore = ficobj.score
winner = ficscore.argmin()
bestfit = m1.mle(data, winner)[0]
y0hat = m1.mode(bestfit, [-4, 4])

# Prints results to screen
print "Simulation %d:" % i
print "Winner/value : %d / %f" % (winner, y0hat)
print "Fit wide   : ", ["%.4f" % j for j in fitwide]
print "FIC score  : ", ["%.4f" % j for j in ficscore]
print "FIC bias   : ", ["%.4f" % j for j in ficobj.sqblist]
print "FIC sterr  : ", ["%.4f" % j for j in ficobj.varlist]
print "FIC ranks  : ", ficobj.ranks, "\n"
```

```
Winner/value : 4 / 1.655969
Fit wide   :  ['0.0270', '1.4443', '-0.1718', '0.8058', '-0.2841', '-0.0957', '↩
    0.2364']
Fit narrow :  ['0.5939', '1.8198']
FIC score  :  ['40.4089', '38.8219', '38.8219', '17.7774', '17.7774', '20.1152']
FIC bias   :  ['40.3679', '38.4896', '38.4896', '13.5676', '13.5676', '0.0000']
FIC sterr  :  ['1.8198', '5.0686', '5.0686', '11.4872', '11.4872', '20.1152']
FIC ranks  :  [6 5 4 2 1 3]
```

In this example the data was generated from test model 2 from (2.3) with sample size $n = 50$. The estimation was done with narrow estimates of ω and J.

The next program does density estimation with wide estimated Average-FIC.

```
# Simulates reality checks for the mode hunting scheme
from numpy import *
from scripts.init import getparam
from scripts import logexp as m1
from scripts import model2 as m2
from scripts import afic

# Gets the correct parameters for test model 5 and generates
# random variables.
mu, tau, p = getparam(model)
data, err = m2.rgen(mu, tau, p, n)

# Fitting the narrow model
m = 5
fitwide = m1.mle(data, m)

# Calculating Jwide, Dn and B.
Jwide = m1.Jwide2(data, fitwide)
yvec = linspace(-10, 10, 10**3)
dvec = matrix(m1.dmu_logdens(yvec, fitwide)).T
B = dvec.T * dvec
Dn = sqrt(n) * fitwide[2:]

# Using the AFIC class to get AFIC scores and determine the winner.
aficobj = afic(Jwide, 2, B, Dn)
ficscore = aficobj.score
winner = ficscore.argmin()
bestfit = m1.mle(data, winner)[0]

# Calculating the mean integrated squared error
xvec = linspace(-10, 10, 10**3)
h = 20.0 / 10**3
yvec = (m1.pdf(xvec, bestfit) - m2.pdf(xvec, mu, tau, p))**2
mise = h * (sum(yvec) - 0.5*(yvec[0] + yvec[-1]))

# Print results to screen
print "Simulation %d:" % i
print "Winner/value : %d / %f" % (winner, mise)
print "AFIC score   : ", ["%.4f" % j for j in ficscore]
print "AFIC sqb     : ", ["%.4f" % j for j in aficobj.sqblist]
print "AFIC stdev   : ", ["%.4f" % j for j in aficobj.varlist]
print "AFIC ranks   : ", aficobj.ranks
```

```
Winner/value : 2 / 0.011152
Fit wide     :  ['0.3349', '1.2937', '-0.2054', '1.1012', '0.2390', '0.0138', '↩
     -0.2273']
AFIC score   :  ['128.7505', '199.9578', '30.9024', '33.1676', '66.5957', '↩
     67.0647']
AFIC bias    :  ['128.7505', '199.9247', '0.0000', '0.0000', '5.5936', '0.0000']
AFIC stdev   :  ['0.0000', '3.6410', '30.9024', '33.1676', '66.3604', '67.0647']
AFIC ranks   :  [5 6 1 2 3 4]
```

In this example the data was generated from test model 2 from (2.3) with sample size $n = 50$. The estimation was done with wide estimates of ω and J.

C.2 Python code for Kernel Estimation

```
from numpy import array, std, mean, sum, min
from numpy import linspace, argmax, ndarray, ones, sqrt, pi
from scipy.stats import norm, scoreatpercentile, skew
import matplotlib.pyplot as plt
from scipy.optimize import fmin

def kernel(y, data, bw='silverman'):
    '''kernel(y, data, bw) -> vector
    Kernel density estimate given data and smoothing parameter bw.'''
    n = len(data)
    if (type(y) != ndarray):
        y = array([y])

    m = len(y)
    s = std(data)

    if bw == 'silverman':
        Q1 = scoreatpercentile(data, 25)
        Q3 = scoreatpercentile(data, 75)
        A = min([s, (Q3 - Q1) / 1.34])
        bw = 0.9*A*n**(-0.2)

    u = (ones((n, m)) * y).T - data
    return sum(norm.pdf(u/bw), axis=1) / (n*bw)

def mode(data):
    '''mode(data) -> value
    Estimates the mode with optimal mode bandwidth.'''
    n = len(data); s = std(data)
    kneg = lambda y: -kernel(y, data, 'silverman')
    grid = linspace(min(data), max(data), int(max(data) - min(data))*50)

    # Estimating  f(y0) and y0first
    fval = kernel(grid, data, 'silverman')
    y0, f1 = fmin(kneg, grid[argmax(fval)], full_output=1, disp=0, xtol=1e-5)[:2]
    f1 = f1*(-1)

    # Estimating the third derivative
    k = lambda y: kernel(array([y]), data, 0.9289 * std(data) * n**(-1.0/11))
    h = 1e-3
    f3 = (-k(y0 - 3*h/2) + 3*k(y0-h/2) - 3*k(y0 + h/2) + k(y0 + 3*h/2)) / (h**3)

    # Final bandwidth and mode estimation
    bwfinal = ((f1 * sqrt(2*pi)**(-1) / (3.0*(f3)**2)) / n)**(1.0/7)
    kneg2 = lambda y: -kernel(y, data, bwfinal)
    fval = kernel(grid, data, bwfinal)

    y0final = fmin(kneg2, grid[argmax(fval)], full_output=0, disp=0, xtol=1e-5)

    return y0final

def mode2(data, bw='silverman'):
    '''mode(data) -> value
    Estimates the mode with Silverman's rule-of-thumb.'''
    kneg = lambda y: -kernel(y, data, bw)
    grid = linspace(min(data), max(data), int(max(data) - min(data))*50)
    fval = kernel(grid, data, bw)
    y0final = fmin(kneg , grid[argmax(fval)], full_output=0, disp=0, xtol=1e-5)

    return y0final
```

C.3 Python code for the Log Expanded Model

```
from scipy.optimize import fmin_powell

from numpy import array, squeeze, reshape, ones, linspace, sqrt, cos
from numpy import nonzero, ceil, mean, std, pi, matrix, concatenate
from numpy import sum, zeros, cov, sin, exp, log, argmin

from hessian import hessian, gradient
from scipy.stats import norm
from scipy.integrate import quad
from init import getparamhome

def pdf(y, p):
    ''' pdf(y, p) -> value
    Probability density function.'''
    try:
        y = array([y]).squeeze().reshape((1,))
    except:
        y = array(y).squeeze()

    n = y.size
    xi = p[0]; sigma = sqrt(p[1]**2)

    if (len(p) == 2):
        a = zeros(1)
    else:
        a = p[2:]

    u = (y - xi)/sigma
    m = len(a)
    j = ones((n, m)) * linspace(1, m, m)

    p1 = norm.pdf(u)/sigma
    p2 = a * (sqrt(2) * cos(j.T * pi * norm.cdf(u))).T

    return p1 * exp(sum(p2, axis=1)) / _km(a)

def _km(a, N=10000):
    '''_km(a) -> value
    Normalizing constant'''
    m = len(a)
    uvec = (linspace(0, 1, N) * ones((m, N))).T
    uvec = exp(sum(a * sqrt(2) * cos(linspace(1, m, m) * pi * uvec), axis=1))
    return N**(-1) * (sum(uvec) - (uvec[0] + uvec[-1])/2)

def loglik(p, *args):
    ''' loglik(p, *args) -> value
    Returns the negative log likelihood. Data vector must be in *args.'''
    dvec = pdf(array(args).squeeze(), p)
    return -sum(log(dvec))

def startvalue(data, m):
    ''' startvalue(data, m) -> vector
    Tries to give a good starting point for optimizing algorithms.'''
    x0 = array([mean(data), std(data)])
    fits = [x0]
    for i in range(1, m+1):
        x0 = concatenate([fits[-1], zeros(1)])
        fits.append(fmin_powell(loglik, x0, args=[data], disp=0))

    return fits

def mle(data, m):
    ''' mle(data, m) -> (fit, log likelihood)
```

```
64      Fits model to data with m parameters in the log expansion.'''
65      if (m == 0):
66          fit = array([mean(data), std(data)])
67          return fit, -loglik(fit, data)
68      else:
69          fit0 = startvalue(data, m)[-1]
70          ll = loglik(fit0, data)
71          augll = lambda p: (loglik(p, data) - ll) * 1e7
72          output = fmin_powell(augll, fit0, disp=0, xtol=1e-10, ftol=1e-5, maxiter↩
                =1000)
73          return output, -loglik(output, data)
74
75  def Jwide(p):
76      ''' Jwide2(data, p) -> matrix
77      Fisher information matrix, evaluated at a = 0'''
78      sigma = p[1]; m = len(p) - 2
79      J = eye(m+2)
80      c, d = zeros(m), zeros(m)
81      for j in range(1, m+1):
82          covfunc1 = lambda u: u * sqrt(2) * cos(j*pi*norm.cdf(u)) * norm.pdf(u)
83          covfunc2 = lambda u: (u**2-1) * sqrt(2) * cos(j*pi*norm.cdf(u)) * norm.↩
                pdf(u)
84
85          c[j-1] = quad(covfunc1, -10, 10, epsabs=1e-10)[0]
86          d[j-1] = quad(covfunc2, -10, 10, epsabs=1e-10)[0]
87
88      J[0,2:] = c/sigma; J[1,2:] = d/sigma
89      J[2:,0] = c/sigma; J[2:,1] = d/sigma
90      J[0, 0] = 1.0/sigma**2
91      J[1, 1] = 2.0/sigma**2
92
93      return J
94
95  def Jwide2(data, p):
96      ''' Jwide2(data, p) -> matrix
97      Empirical hessian of the log likelihood function'''
98      return hessian(loglik, p, args=data, h=1e-4) / len(data)
99
100 def Kwide(data, p):
101     '''Kwide(data, p) -> matrix
102     Empirical covariance matrix of the score function'''
103     return cov(dmu_logdens(data, p))
104
105 def mode(p, I):
106     ''' mode(xi, sigma, a, I) -> vector
107     Gives the global maximum of f given a set of parameters.'''
108     fneg = lambda y: -pdf(y, p)
109     xvec = linspace(I[0], I[1], 200)
110     start = xvec[argmin(fneg(xvec))]
111     return fmin_powell(fneg, start, full_output=1, disp=0, xtol=1e-8)[0]
112
113 def dmu_mode(y, p):
114     ''' dmu_mode(y, p) -> value
115     The derivative of the mode, with respect to p'''
116     n = y.size
117     a = p[2:]
118     m = len(p) - 2
119     j = linspace(1, m, m)
120     u1 = (((y - p[0])/p[1]) * ones((m, n))).T
121     u2 = (y - p[0]) / p[1]
122     h = 1e-4
123
124     phi = norm.pdf
125     Phi = norm.cdf
126     dphi = lambda v: (phi(v+h) - phi(v-h))/(2*h)
127
128     P = sum(a*j*sin(j*pi*Phi(u1)), axis=1)
```

```
129     denom = 1/sqrt(2) + pi*dphi(u2)*P + (pi*phi(u2))**2 * sum(a*j**2*cos(j*pi*↩
            Phi(u1)), axis=1)
130     du = ones((m+2, n))
131     du[1] = u2
132     for k in range(1, m+1):
133         du[k+1] = -k*p[1]*pi*phi(u2)*sin(k*pi*Phi(u2)) / denom
134 
135     return du
136 
137 def dmu_dens(y, p):
138     ''' dmu_mode(y, p) -> value
139     The derivative of the density function at a point y, with respect to p'''
140     n = y.size
141     a = p[2:]
142     m = len(p) - 2
143     k = linspace(1, m, m)
144     du = zeros((m+2, n))
145 
146     f = lambda p: pdf(y, p)
147     for k in range(m+2):
148         h = zeros(m+2); h[k] = 1e-4
149         du[k] = (f(p+h) - f(p-h)) / sum(2*h)
150 
151     return du
152 
153 def dmu_logdens(y, p):
154     ''' dmu_mode(y, p) -> value
155     The derivative of the log density function at a point y, with respect to p↩
            '''
156     return dmu_dens(y, p) / pdf(y, p)
157 
158 def rgen(p, n):
159     ''' rgen(p, n) -> value
160     Random number generator (acceptance-rejection algorithm)'''
161     Phi = norm.pdf
162     c = exp(sqrt(2)*sum(abs(p[2:]))) / _km(p[2:])
163 
164     x = random.normal(p[0], p[1], 2*ceil(c)*n)
165     u = random.uniform(0, c*norm.pdf(x), 2*ceil(c)*n)
166     b = x[nonzero([u < pdf(x, p)][0] * x)]
167 
168     return b[:n]
```

C.4 Python code for the Normal Mixture

```
from numpy import ones, ndarray, array, matrix, inf, sum, abs, linspace
from numpy import *#random, min, max, sqrt

from scipy.stats import norm
from scipy.optimize import fmin
from scipy.integrate import quad

def pdf(y, mu, tau, p):
    '''pdf(y, mu, tau, p) -> value
    Probability density function '''
    if (type(y) != ndarray):
        y = array([y])

    n = len(y); k = len(mu)
    y = (y * ones((k, n))).T
    u = norm.pdf((y - mu) / tau) / tau
    return sum(p * u, axis=1)

def cdf(y, mu, tau, p):
    '''cdf(y, mu, tau, p) -> value
    Cumulative distribution function '''
    n = len(y); k = len(mu)
    y = (y * ones((k, n))).T
    u = norm.cdf((y - mu) / tau)
    return sum(p*u, axis=1)

def ppf(u, mu, tau, p):
    '''ppf(u, mu, tau, p) -> value
    Probability point function (inverse cumulative)'''
    up = u*(max(mu) - min(mu)) + min(mu)
    err = inf
    while (err > 1e-5):
        un = up - (cdf(up, mu, tau, p) - u)/pdf(up, mu, tau, p)
        err = sum(abs(up-un))
        up = un
    return un, err

def mode(mu, tau, p, I):
    '''mode(xi, sigma, a, I) -> vector
    Gives the maximum of the pdf at interval I given parameters.'''
    fneg = lambda y: -pdf(y, mu, tau, p)
    xvec = linspace(I[0], I[1], 200)
    start = xvec[argmin(fneg(xvec))]
    return fmin(fneg, start, full_output=1, disp=0, xtol=1e-8)[0]

def rgen(mu, tau, p, n):
    '''rgen(mu, tau, p, n) -> vector
    Generates n random variables given parameters with the composite method.'''
    k = len(mu)
    pos = random.multinomial(1, p, n)
    r = random.normal(0, 1, n)

    return sum(((r * ones((k, n))).T * tau + mu) * pos, axis=1)

def moments(mu, tau, p):
    '''moments(mu, tau, p) -> vector
    Expected value, variance, skewness and the exess kurtosis '''
    ex1 = sum(p*mu)
    ex2 = sum(p * (mu**2 + tau**2))
    ex3 = sum(p * (mu**3 + 3*mu * tau**2))
    ex4 = sum(p * (mu**4 + 6*mu**2 * tau**2 + 3*tau**4))

    variance = ex2 - ex1**2
```

```
64      skewness = (ex3 - 3*ex1*variance - ex1**3) / variance**(3./2)
65      kurtosis = (ex4 - 4*ex3*ex1 + 6*ex2*ex1**2 - 3*ex1**4) / variance**2 - 3
66
67      return ex1, variance, skewness, kurtosis
68
69  def kernelrmse(mu, tau, p, n):
70      '''kernelrmse(mu, tau, p, n) -> value
71      The approximate variance of the kernel mode estimate'''
72      y0 = mode(mu, tau, p, [-4, 4])
73      h  = 1e-4
74      ft  = lambda y: pdf(array([y]), mu, tau, p)
75      d2f = lambda y: (ft(y - 3*h/2) - ft(y - h/2) - ft(y + h/2) + ft(y + 3*h/2)) ↩
            / (2*h**2)
76      d3f = lambda y: (-ft(y - 3*h/2) + 3*ft(y-h/2) - 3*ft(y + h/2) + ft(y + 3*h↩
            /2)) / h**3
77
78      C = ft(y0) * sqrt(2*pi)**(-1) / (3 * (d3f(y0)**2))
79      bw = (C/n)**(1.0/7)
80
81      if (bw == inf):
82          print 'lol'
83          return sqrt(moments(mu, tau, p)[1] / n)
84      else:
85          V = (2*sqrt(2*pi))**(-1)
86          variance = ft(y0) * V / (n * bw**3 * d2f(y0)**2)
87          bsq = (bw**2 * d3f(y0) / (2*d2f(y0)))**2
88          return sqrt(variance + bsq)
89
90  def kernelmise(mu, tau, p, n):
91      '''kernelmise(mu, tau, p, n) -> value
92      The approximate mise of the kernel density estimate'''
93      s = moments(mu, tau, p)[1]
94      Q1 = ppf(array([0.25]), mu, tau, p)[0]
95      Q3 = ppf(array([0.75]), mu, tau, p)[0]
96      bw = 0.9 * min(sqrt(s), (Q3 - Q1)/1.34) * n**(-1.0/5)
97      h = 1e-4
98      g = lambda y: pdf(y, mu, tau, p)
99      d2g = lambda y: (g(y-3*h/2) - g(y-h/2) - g(y+h/2) + g(y+3*h/2)) / (2*h**2)
100
101     variance = 1/(2*sqrt(pi)*n*bw)
102
103     n = 10**3
104     xvec = linspace(-20, 20, n)
105     h = 20.0 / n
106     y4 = d2g(xvec)**2
107     bias = bw**4 * (sum(y4) - (y4[0] + y4[-1])/2) * h / 4
108
109     return bias + variance
```

C.5 Python code for the FIC class

```
from numpy import matrix, array, sqrt, zeros, eye, sign
from numpy.linalg import inv

class fic:
    ''' Class for calculating FIC scores for an arbitrary model selection problem. Theory
    for this program is taken from Claesken & Hjort 2008.

    ficobj = fic(Jwide, ddtheta, ddgamma, Dn)'''
    def __init__(self, Jwide, ddtheta, ddgamma, Dn):
        self.Jwide = matrix(Jwide)
        J = self.Jwide
        p = len(ddtheta); self.p = p
        q = len(ddgamma); self.q = q
        ddtheta = matrix(ddtheta).T; self.ddtheta = ddtheta
        ddgamma = matrix(ddgamma).T; self.ddgamma = ddgamma
        self.Dn = matrix(Dn).T

        self.omega = J[p:,:p] * inv(J[:p,:p]) * ddtheta - ddgamma
        self.tau0 = sqrt(ddtheta.T * inv(J[:p, :p]) * ddtheta)
        self.Q = inv(J)[p:, p:]
        self._runfic()

    def _runfic(self):
        q = self.q
        ficscore, biaslist, stdlist = zeros(q+1), zeros(q+1), zeros(q+1)

        # For narrow model
        sqb = self.omega.T * (self.Dn * self.Dn.T - self.Q) * self.omega
        stdlist[0] = self.tau0
        ficscore[0] = sqrt(self.tau0**2 + max(sqb, 0))
        biaslist[0] = sign(sqb) * sqrt(abs(sqb))

        # For m = 1...
        for k in range(1, q+1):
            stdlist[k], biaslist[k], ficscore[k] = self.calculate(range(1, k+1))

        self.ficscore = ficscore; self.stdlist = stdlist; self.biaslist = biaslist
        self.ranks = ficscore.argsort().argsort() + 1

    def calculate(self, subset):
        ''' calculate(subset) -> value
        Fic score for submodel defined by list of parameters to include.'''
        p = self.p; q = self.q
        pi = eye(q)[array(subset) - 1]
        J = self.Jwide
        Dn = self.Dn
        ddgamma = self.ddgamma
        ddtheta = self.ddtheta

        omega = self.omega
        tau0 = self.tau0
        Q = self.Q
        Iq = eye(q)

        Qs = inv(pi * inv(Q) * pi.T)
        Gs = pi.T * Qs * pi * inv(Q)
        v = sum( tau0**2 + (pi * omega).T * Qs * (pi * omega) )
        sqb = sum( omega.T * (Iq - Gs) * (Dn * Dn.T - Q) * (Iq - Gs).T * omega )
        score = sqrt(v + max(sqb, 0))

        return sqrt(v), sign(sqb) * sqrt(abs(sqb)), score
```

C.6 Python code for the AFIC class

```
from numpy import matrix, array, sqrt, zeros, trace, eye, sign, max, sum
from numpy.linalg import inv

class afic:
    ''' Class for calculating Average–FIC scores for an arbitrary model ↩
        selection problem. Theory
    for this program is taken from Claesken & Hjort 2008 section 6.9.

    ficobj = afic(Jwide, p, B, Dn)
    '''
    def __init__(self, Jwide, p, B, Dn):
        self.Jwide = matrix(Jwide)
        self.p = p; self.q = len(Jwide) - p
        self.Dn = Dn; self.B = B
        self._A(); self._runfic()

    def _A(self):
        p = self.p; q = self.q; B = self.B
        J00 = matrix(self.Jwide[:p, :p])
        J01 = matrix(self.Jwide[:p, p:]); J10 = J01.T
        J11 = matrix(self.Jwide[p:, p:])
        self.A = J10*inv(J00)*B[:p,:p]*inv(J00)*J01 - J10*inv(J00)*B[:p ,p:] - B↩
            [p:,:p]*inv(J00)*J01 + B[p:,p:]
        self.Q = inv(self.Jwide)[p:,p:]

    def _runfic(self):
        J = self.Jwide
        Dn = matrix(self.Dn).T
        p = self.p; q = self.q
        Q = self.Q

        ficscore, biaslist, stdlist = zeros(q+1), zeros(q+1), zeros(q+1)

        # For narrow model
        sqb1 = trace((Dn * Dn.T - Q) * self.A)
        biaslist[0] = sign(sqb1) * sqrt(abs(sqb1))
        ficscore[0] = sqrt((sqb1 > 0) * abs(sqb1))

        # For m = 1...
        for k in range(1, q+1):
            stdlist[k], biaslist[k], ficscore[k] = self.calculate(range(1, k+1))

        self.score = ficscore
        self.biaslist = biaslist
        self.stdlist = stdlist
        order = self.score.argsort()
        self.ranks = order.argsort() + 1

    def calculate(self, subset):
        p = self.p; q = self.q
        pi = matrix(eye(q)[array(subset) - 1])
        Iq = matrix(eye(q))
        Q = self.Q; Dn = self.Dn
        Qs = inv(pi * inv(Q) * pi.T)
        Gs = pi.T * Qs * pi * inv(Q)
        v  = sum( trace(pi.T * Qs * pi * self.A) )
        sqb1 = sum( trace((Iq - Gs) * (Dn * Dn.T - Q) * (Iq - Gs).T * self.A) )
        sqb2 = max(sqb1, 0)

        return sqrt(v), sign(sqb1)*sqrt(abs(sqb1)), sqrt(v + sqb2)
```

Bibliography

H. Akaike. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, 1(1):267–281, 1973.

A. R. Barron and C.-H. Sheu. Approximations of density functions by sequences of exponential estimates. *Annals of Statistics*, 19:1347–1369, 1991.

G. Claeskens and N. L. Hjort. The focused information criterion. *Journal of the American Statistical Association*, 98(464):900–916, 2003.

G. Claeskens and N. L. Hjort. Goodness of fit via nonparametric likelihood ratios. *Scandinavian Journal of Statistcs*, 31:487–513, 2004.

G. Claeskens and N. L. Hjort. *Model Selection and Model Averaging*. Cambridge University Press, 2008.

D. R. Cox. Notes on analysis of mixed frequency analysis. *British Journal of Mathematical and Statistical Psychology*, 19:39–47, 1966.

G. Dahlquist and A. Bjorck. *Numerical Methods in Scientific Computing vol. I*. siam, 2008.

W. F. Eddy. Optimum kernel estimators of the mode. *The Annals of Statistics*, 8 (4):870–882, 1980.

W. F. Eddy. Asymptotic distributions of kernel estimators of the mode. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 59(3):279–290, 1982.

D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons Inc., 1999.

P. J. Huber. The behavior of maximum likelihood estimates under nonstandard conditions. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:221–233, 1967.

K. Knight. *Mathematical Statistics*. Chapman and Hall/CRC, 2000.

H. P. Langtangen. *A Primer on Scientific Programming with Python*. Springer-Verlag Berlin, 2010.

E. L. Lehmann. *Elements of Large-Sample Theory*. Springer-Verlag New York, 1999.

E. A. Nadaraya. On nonparametric estimates of density functions and retgression curves. *Theor. Probability Appl.*, 10:186–190, 1965.

E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, Sep 1962.

M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, 27:832–837, 1956.

R. Y. Rubenstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. Wiley, 2008.

B. W. Silverman. *Density Estimation*. Chapman and Hall, 1986.

G. Teschl. *Topics in Real and Functional Analysis*. http://www.mat.univie.ac.at/ gerald/ftp/book-fa/, 2011.

J. Van Ryzin. On strong consistency of density estimates. *Ann. Math. Statist.*, 40:1765–1772, 1969.